UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE CIENCIAS MATEMÁTICAS

E.A.P. DE ESTADÍSTICA

Método de clasificación para evaluar el riesgo crediticio:

una comparación

TESIS

para optar el título profesional de Licenciada en Estadística

AUTORA

Geraldine Judith Vigo Chacón

ASESORA

Emma Cambillo M.

Lima – Perú

2010

La falta mas grave es no tener conciencia de nuestra propia falta.

Al verdadero Dios que es quien ilumina nuestros caminos.

Y a quien verdaderamente comparte a nuestro lado estos caminos.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a mi asesora Mg. Emma Cambillo M. por su gran apoyo, predisposición y dedicación en el desarrollo de la presente tesis.

También deseo expresar mi agradecimiento a todos los docentes de la Facultad de Ciencias Matemáticas de la UNMSM por sus enseñanzas.

Finalmente mi agradecimiento a la UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS, por ser mi casa de estudios.

ÍNDICE

RESUMEN

Se comparan dos métodos clásicos de clasificación: Análisis de Regresión Logística y Árboles de Clasificación, con el método de Redes Neuronales. La comparación se realizó en base al poder de clasificación y predicción de los modelos obtenidos en la evaluación del Riesgo Crediticio, siendo Redes Neuronales el mejor método por tener mayor poder de clasificación y predicción. Para el análisis se utilizó una Base de Datos de Riesgo Crediticio. Asimismo, se establecen las ventajas y desventajas en el empleo de cada método.

Palabras Claves: *Análisis de Regresión Logística, Árboles de Clasificación, Redes Neuronales.*

ABSTRACT

Two classics methods of classification are compared: Analysis of Logistic Regression and Classification Trees with the method of Neural Networks. The comparison realized through his power of classification and prediction of the models obtains in the evaluation of credit risk, Neural Networks is the best method, because it has high power of classification and prediction. For the analysis used a database of credit risk. Likewise found the advantages and disadvantages in the use of each method.

Keys Words: *Analysis Logistic Regression, Classification Trees, Neural Networks.*

INTRODUCCIÓN

Uno de los problemas en la actividad científica, académica y financiera es obtener un buen método de clasificación, es decir poder etiquetar a un sujeto u objeto de acuerdo a ciertas características; además cuando se clasifica a un sujeto en un grupo determinado, a partir de los valores de una serie de parámetros medidos u observados, y esa clasificación tiene un cierto grado de error, resulta razonable pensar en la utilización de una metodología probabilística, que nos permita cuantificar ese error, sin embargo el mejor método debe tener una menor tasa de error en la clasificación, y también conocer cuando es recomendable la aplicación de cada uno de los métodos en este tipo de análisis.

Los métodos de clasificación tienen como objetivo identificar el grupo al cual un sujeto u objeto pertenece y explicar las relaciones que influyen en el grupo en donde se encuentra un objeto. En estos métodos la variable dependiente debe ser de escala no métrica.

Entre los métodos de clasificación mas conocidos están el Análisis de Regresión Logística, Árboles de Clasificación y Análisis Discriminante. Como métodos alternativos están Redes Neuronales, Máquinas Vectoriales de Soporte y Algoritmos Genéticos. Estos métodos sirven para la construcción de

modelos de *scoring* crediticio, dichos modelos permiten determinar si un individuo esta en capacidad de cumplir con las exigencias de un crédito.

La importancia de esta problemática radica principalmente en encontrar un método adecuado, el cual tenga una menor tasa de error en la clasificación; además es de interés conocer cuando es recomendable la aplicación de cada método en el análisis del scoring crediticio u otro.

En algunos casos los métodos de clasificación han sido combinados, por ejemplo: la regresión logística y las máquinas vectoriales de soporte fueron combinados con la finalidad de obtener un modelo con un alto poder de discriminación y legibilidad. En otros casos las entidades han hecho uso de las técnicas del Data Mining, por ejemplo: En España se llevó a cabo un proyecto denominado MINERVA, el cual trataba de detectar fraude en tarjetas de crédito, en ese proyecto se aplicó el método de Redes Neuronales para la detección del fraude obteniéndose resultados satisfactorios. Respecto a la detección de patrones de morosidad el Análisis de Regresión Logística y Árboles de Clasificación son los más utilizados.

Los objetivos del presente trabajo son:

- Encontrar un método de clasificación adecuado que represente a la variable riesgo crediticio, con la finalidad de predecir qué personas de las que solicitan crédito son un riesgo y que este método tenga un menor error en la clasificación.
- Establecer una comparación entre los métodos Análisis de Regresión Logística, Árboles de Clasificación y Redes Neuronales para determinar al mejor en clasificación y predicción en la evaluación del riesgo crediticio.
- Conocer las ventajas y desventajas en la aplicación de cada uno de los métodos.

En el capítulo I se analizó el método de Análisis de Regresión Logística. La Regresión Logística forma parte de los Modelos Lineales Generalizados, donde la función de enlace es la función logit. La regresión logística se estima de forma similar a la regresión múltiple estimándose primero un modelo base, sin embargo la media no se emplea en el cálculo de la suma total de cuadrados sino para el valor del logaritmo de la verosimilitud.

En el capítulo II se analizó el método de Árboles de Clasificación. El cual es utilizado en el análisis estadístico y minería de datos, para la clasificación y predicción. Los árboles de clasificación también llamados de decisión o de identificación, son particiones secuenciales del conjunto de datos para maximizar las diferencias de la variable dependiente.

En el capítulo III se analizó el método de Redes Neuronales, un método más asociado con la extracción de los datos, que intenta aprender mediante ensayos repetidos como organizarse mejor a si misma para conseguir maximizar la predicción. Las redes neuronales (RN) pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos a priori ni especificar funciones de distribución de probabilidad. Las RN son sistemas dinámicos y adaptables; son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones, y son adaptables debido a la capacidad de ajuste de los elementos (neuronas) que componen el sistema. En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtenga ciertos resultados específicos.

En el capítulo IV se realizó una comparación de los modelos obtenidos mediante estos tres métodos, determinando cual de los métodos es el mejor en base a su poder de clasificación y predicción. Para realizar el análisis de los métodos se utilizó una Base de Datos de Riesgo Crediticio.

CAPÍTULO I:

Análisis de Regresión Logística

1.1 Introducción

El análisis de regresión logística es una técnica para el estudio de la relación entre una o mas variables independientes y una variable dependiente de tipo dicotómica, representa la ocurrencia o no de un suceso, por ejemplo: un paciente muere o no antes del alta, una persona deja o no de fumar después de un tratamiento, una persona vota a favor o no de un candidato, una persona compra artículos o no de una determinada marca, etc.

Un modelo de regresión logística permite estimar o predecir la probabilidad de que un individuo posea una característica en función de una determinada o unas determinadas características individuales. La regresión logística forma parte de los modelos lineales generalizados, donde la función de enlace es la función logit. Este modelo comúnmente presenta una forma de "S", limitada en el eje de las ordenadas entre los valores 0 y 1. El modelo antes descrito se denomina ***"Función Logística"*.**

En 1937, Bartlett utilizó la transformación *log[y/(1-y)]* para analizar proporciones. Fisher y Yates sugirieron en 1938 el uso de esa transformación para analizar datos binarios. El término ***logit*** fue introducido por Joseph Berkson en 1944 para designar esta transformación y sus trabajos incentivaron la utilización de la regresión logística. Jerome Cornfield utilizó este método para el cálculo de **ODDS RATIO** como valores aproximados del riesgo relativo en estudios de casos y controles. [1]

Sin embargo el principal difusor de la regresión logística fue David R. Cox en 1970 con su libro "*The Analysis of Binary Data*". La introducción de los modelos lineales generalizados (GLM), por los estadísticos británicos John Nelder y R.W.M. Wedderburn, en 1972, unifican toda la teoría existente en cuanto a modelos probit y modelos logísticos, con los modelos lineales basados en la distribución normal, así como el análisis de la varianza. El principal algoritmo para ajustar estos modelos se denomina "*Fisher scoring*", debido a que fue introducido por Fisher en 1935 para ajustar modelos probit de máxima verosimilitud. [1]

El objetivo fundamental de la regresión logística es determinar si hay relación entre una variable predicha o variable respuesta y un conjunto de predictores o variables regresoras. La regresión logística permite modelar cómo influye en la probabilidad de aparición de un suceso (generalmente de una variable dicotómica) la presencia o no de diversos factores.

En este análisis es conveniente tener en cuenta de que las variables categóricas deben ser codificadas de forma apropiada. [2]

1.2 Definición

Sean $\mathbf{X_1, X_2,\ldots, X_k}$ un conjunto de variables regresoras.

Sea **Y** una variable dependiente dicotómica, que toma los valores 0 y 1.

Se busca determinar $p = P\ Y = 1/\ X_1, X_2, ..., X_k$, donde p es la probabilidad de éxito.

Se construye un modelo de la forma:

$$P\ Y = 1/\ X_1, X_2, ..., X_k\ = p\ \ X_1, X_2, ..., X_k; \beta \qquad \text{(I.1)}$$

donde $p\ \ X_1, X_2, ..., X_k; \beta$, es una función que recibe el nombre de función de enlace (función de probabilidad) cuyo valor depende de un vector de parámetros $\beta = (\beta_1, \beta_2, ..., \beta_k)'$.

1.3 Modelo de la Regresión Logística

La variable dependiente ***Y*** toma valor 1 si ocurre el suceso, y valor 0 si no ocurre el suceso. Por otra parte interesa estudiar la relación entre una o más variables independientes o explicativas: X_1, X_2,..., X_k y la variable Y. El modelo logístico establece la siguiente relación entre ***la probabilidad de que ocurra el suceso***, dado que el individuo presenta los valores $X_1=x_1$, $X_2 = x_2$,...,$X_k = x_k$.

En este caso la ecuación de regresión tiene que ser diferente de la que se emplea en regresión múltiple ($\mathrm{Y}=\beta'\mathrm{X}+\varepsilon$). En este modelo lo que se predice no es directamente la variable sino ***la probabilidad*** de que la variable adopte cierto valor. La variable dependiente es pues una probabilidad. Para predecir una probabilidad pueden utilizarse diferentes funciones, entre las que destaca la logística. Esta función es la base del cálculo de la probabilidad de p que queremos predecir.

Si llamamos X_i a los predictores, sea:

$$p = p\ \ X_1, X_2, ..., X_k; \beta\ \ = G\ \ \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k\ \ = \mathrm{G}\ \ \mathrm{X}'\beta \qquad \text{(I.2)}$$

donde la función de densidad acumulada de la función logística, la cual es usualmente denotada como:

$$\log\left(\frac{p\ \ X_1, X_2, ..., X_k; \beta}{1 - p\ \ X_1, X_2, ..., X_k; \beta}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k \qquad \text{(I.3)}$$

es llamado modelo ***logit***. Esto indica que existe una relación lineal entre el cociente de probabilidades (la probabilidad de pertenecer a un grupo (éxito) dividido por la probabilidad de pertenecer al otro (fracaso)) y los predictores.

Si se aplica la exponencial a la expresión (I.3), la ecuación queda expresada de la siguiente manera:

$$p = \frac{e^{X'\beta}}{1 + e^{X'\beta}} = \frac{1}{1 + e^{-X'\beta}} \qquad \text{(I.4)}$$

La ecuación anterior tiene la forma de la ecuación de regresión múltiple, donde β_0 es la constante y los β_i son los coeficientes de los predictores X_i correspondientes.

Los supuestos del modelo son:

- Ausencia de colinealidad entre las variables regresoras.
- Los errores tienen distribución binomial.
- No linealidad de la variable de respuesta.

El modelo de regresión logística es robusto con respecto al incumplimiento del supuesto de igualdad de las matrices de covarianza entre grupos.

Los supuestos esenciales de la regresión logística son:

- Independencia entre las observaciones sucesivas.
- Existencia de una relación lineal entre logit(x) y los predictores X_1,..., X_k.

1.4 Estimación del Modelo

1.4.1. Estimación de los coeficientes

Aunque existen otros métodos, el más empleado es el de máxima verosimilitud, que consiste en maximizar la función de verosimilitud de la muestra.

Para ello se considerará una muestra aleatoria simple de tamaño n dada de la siguiente forma: $X_i^{'}, Y_i \;; i = 1,2,...,n$, donde: X_i son los valores de las variables independientes del i-ésimo individuo de la muestra, $Y_i = 0,1$ es el valor observado de la variable dependiente del i-ésimo individuo de la muestra.

Además $Y / \; X_1, X_2, ..., X_k \sim Binomial \; 1, p \;\; Y = 1 / \; X_1, X_2, ..., X_k \;\; ; \beta$, y el número de éxito en *n* repeticiones tiene una distribución binomial *B(n, p)*. La función de verosimilitud es:

$$\ell(\beta) = \mathrm{L} \;\; \beta / \;\; \mathrm{X}_1^{'}, \mathrm{y}_1 \;\; ,..., \;\; \mathrm{X}_\mathrm{n}^{'}, \mathrm{y}_\mathrm{n} \;\; = \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{p}_\mathrm{i}^{\mathrm{y}_\mathrm{i}} (1 - \mathrm{p}_\mathrm{i})^{1-\mathrm{y}_\mathrm{i}} \qquad \text{(I.5)}$$

donde $p_i = \dfrac{e^{X_i^{'}\beta}}{1 + e^{X_i^{'}\beta}}; \quad i = 1,2,....,n$.

Aplicando logaritmo neperiano a ambas expresiones, se tiene:

$$L = \ln\ \ell(\beta)\ = \sum_{i=1}^{n}\left(y_i x_i^{'}\beta + \ln\left(\frac{1}{1+e^{x_i^{'}\beta}}\right)\right) \qquad \text{(I.6)}$$

El vector de parámetros β se estima mediante el método de máxima verosimilitud. Una forma de calcularla es bajo la *estimación incondicional*, la cual maximiza la función de verosimilitud anterior, para esto la ecuación (I.6) se deriva y se iguala al valor cero:

Por β_0:

$$\frac{\partial}{\partial\beta_0} L = 0$$

$$\Rightarrow \sum_{i=1}^{n}\left(y_i - \frac{e^{x_i^{'}\hat{\beta}}}{1+e^{x_i^{'}\hat{\beta}}}\right) = 0$$

$$\Rightarrow \sum_{i=1}^{n}\ y_i - \hat{p}_i\ = 0 \qquad \text{(I.7)}$$

Y por β_j:

$$\frac{\partial}{\partial\beta_j} L = 0$$

$$\Rightarrow \sum_{i=1}^{n} x_{ij}\ \ y_i - \hat{p}_i\ = 0 \qquad \text{(I.8)}$$

Las *k+1* ecuaciones (I.7 y I.8) se resuelven mediante métodos iterativos. Este procedimiento es matemáticamente complejo, el proceso es iterativo, es decir se dan a los coeficientes unos valores arbitrarios (habitualmente, aunque no necesariamente el valor 0). La solución final no depende de estos valores pero sí del tiempo de cálculo.

1.4.2. Estimación de la Matriz de Covarianzas de los estimadores de los coeficientes

Para la estimación de la matriz de covarianzas de los coeficientes estimados se debe hallar la segunda derivada parcial del logaritmo de la función de máxima verosimilitud (**L**). [3]

Derivando por β_j:

$$\frac{\partial^2}{\partial\beta_j^2} L = -\sum_{i=1}^{n} x_{ij}^2 p_i \ (1-p_i) \tag{I.9}$$

Derivando por β_j y β_l:

$$\frac{\partial^2}{\partial\beta_j \partial\beta_l} L = -\sum_{i=1}^{n} x_{ij} x_{il} p_i \ (1-p_i) \tag{I.10}$$

Esto es para $j \neq l = 0, 1, \ldots, k$.

Se define la matriz $I(\hat{\beta})$ la cual esta conformada por los valores negativos de las ecuaciones (I.9 y I.10), teniendo la siguiente forma:

$$I(\hat{\beta}) = \begin{bmatrix} \sum_{i=1}^{n} p_i (1-p_i) & \sum_{i=1}^{n} x_{i1} p_i (1-p_i) & \sum_{i=1}^{n} x_{i2} p_i (1-p_i) & \cdots & \sum_{i=1}^{n} x_{ik} p_i (1-p_i) \\ \sum_{i=1}^{n} x_{i1} p_i (1-p_i) & \sum_{i=1}^{n} x_{i1}^2 p_i (1-p_i) & \sum_{i=1}^{n} x_{i1} x_{i2} p_i (1-p_i) & \cdots & \sum_{i=1}^{n} x_{i1} x_{ik} p_i (1-p_i) \\ \sum_{i=1}^{n} x_{i2} p_i (1-p_i) & \sum_{i=1}^{n} x_{i2} x_{i1} p_i (1-p_i) & \sum_{i=1}^{n} x_{i2}^2 p_i (1-p_i) & \cdots & \sum_{i=1}^{n} x_{i2} x_{ik} p_i (1-p_i) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_{ik} p_i (1-p_i) & \sum_{i=1}^{n} x_{ik} x_{i1} p_i (1-p_i) & \sum_{i=1}^{n} x_{ik} x_{i2} p_i (1-p_i) & \cdots & \sum_{i=1}^{n} x_{ik}^2 p_i (1-p_i) \end{bmatrix}$$

Expresándolo en forma matricial se tiene:

$$I(\hat{\beta}) = X'_{(k+1)xn} \Gamma_{nxn} X_{nx(k+1)} \tag{I.11}$$

la cual es de orden $(k+1)x(k+1)$.

La matriz de covarianzas de los coeficientes estimados es obtenida a través de la matriz inversa de $I(\hat{\beta})$, es decir:

$$\text{Var}(\hat{\beta}) = I^{-1}(\hat{\beta}) \tag{I.12}$$

En la diagonal de esta matriz se encuentran las varianzas de $\hat{\beta}$, es decir en el j-ésimo elemento de la diagonal se encuentra la varianza de $\hat{\beta}_j$, los elementos que se encuentran fuera de la diagonal son las covarianzas ($\text{cov}\ \hat{\beta}_j; \hat{\beta}_l$, Para $j \neq l = 0, 1, \ldots, k$).

Para conocer el estimador de la varianza de $\hat{\beta}$ debemos hallar el estimador de $\mathrm{I}\,\hat{\beta}$, dado que la matriz X esta conformado por los datos observados de los individuos, por lo que se deberá determinar el estimador de la matriz Γ cuyos elementos son $p_i(1-p_i) \quad \forall i=1,2,....,n$ y el estimador de p_i es $\hat{p}_i$ [3], entonces:

$$\hat{\Gamma}=\mathrm{V}=\begin{bmatrix} \hat{p}_1\ 1-\hat{p}_1 & 0 & 0 & \cdots & 0 \\ 0 & \hat{p}_2\ 1-\hat{p}_2 & 0 & \cdots & 0 \\ 0 & 0 & \hat{p}_3\ 1-\hat{p}_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \hat{p}_n\ 1-\hat{p}_n \end{bmatrix}. \qquad \text{(I.13)}$$

Luego el estimador de la varianza de $\hat{\beta}$ es:

$$\hat{\mathrm{Var}}\ \hat{\beta}\ =\hat{\mathrm{I}}^{-1}\ \hat{\beta}\ =\ \mathrm{X'VX} \qquad \text{(I.14)}$$

1.5 Evaluación del Modelo

1.5.1 Evaluación de la Significancia del Modelo

Para la evaluación de la significancia del modelo, es decir determinar si las variables independientes son significativas o no, se plantea las siguientes hipótesis:

$$H_0: \beta_1=\beta_2=\beta_3=\beta_4=....=\beta_k=0$$
$$H_1: \textit{Por lo menos un}\ \beta_i\neq 0 \quad \forall i=1,2,...,k$$

con un nivel α de significación.

Estadístico de Prueba:

El estadístico que se plantea para la evaluación de la significancia del modelo es la diferencia del valor de la desvianza del modelo sólo con la constante y del modelo incluyendo las variables independientes, este estadístico sigue una distribución Chi – Cuadrada con k grados de libertad [3], se tiene:

$$G = D(\text{modelo sin variables}) - D(\text{modelo con variables})$$
$$= -2\ln\left(\frac{\text{verosimilitud del modelo sin variables}}{\text{verosimilitud del modelo con variables}}\right) \quad \text{(I.15)}$$

donde:

$$-2\ln(\text{Verosimilitud del modelo sin variables}) = -2\left[n_1 \ln n_1 + n_0 \ln n_0 - n \ln n\right] \quad \text{(I.16)}$$

$$-2\ln(\text{Verosimilitud del modelo con variables}) = -2\sum_{i=1}^{n}\left[y_i \ln \hat{p}_i + (1-y_i)\ln 1-\hat{p}_i\right] \quad \text{(I.17)}$$

Reemplazando las ecuaciones (I.16) y (I.17) en la ecuación (I.15), se obtiene:

$$G = -2\left\{ n_1 \ln(n_1) + n_0 \ln(n_0) - n\ln(n) - \sum_{i=1}^{n} y_i \ln(\hat{p}_i) + (1-y_i)\ln(1-\hat{p}_i) \right\} \sim \chi^2_k$$

Criterio de Decisión:

Se rechaza H_0, si $G > a$

Entonces una o más de las variables Independientes consideradas en el modelo son significativos.

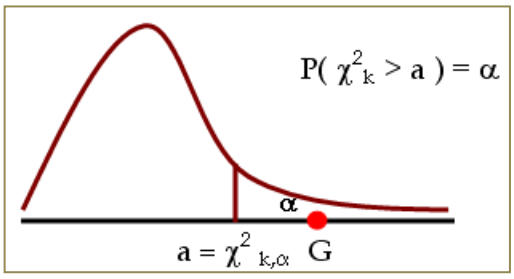


Figura I.18. *Representación gráfica de la región de rechazo de la evaluación del modelo*

1.5.2 Evaluación de la Bondad de Ajuste del Modelo

Desvianza del Modelo (D):

Para la evaluar la bondad de ajuste del modelo, es decir determinar si el modelo ajustado es el adecuado o no, se plantea las siguientes hipótesis:

H_0 : *El modelo ajustado es significativo*

H_1 : *El modelo ajustado no es significativo*

con un nivel α de significación.

Estadístico de Prueba:

La desvianza es la medida del grado de diferencia entre las frecuencias predichas y las observadas del modelo, el mejor modelo será aquel que tenga menor desvianza, este estadístico sigue una distribución Chi – Cuadrada con n – (k+1) grados de libertad [3], se tiene:

$$D = -2\ln\left(\frac{Verosimilitud\ del\ modelo\ ajustado}{Verosimilitud\ del\ modelo\ saturado}\right) \qquad \text{(I.19)}$$

Donde:

$$Verosimilitud\ del\ modelo\ ajustado = \prod_{i=1}^{n} \hat{p}_i^{\,y_i}\ \ 1-\hat{p}_i\ ^{(1-y_i)} \qquad \text{(I.20)}$$

$$Verosimilitud\ del\ modelo\ saturado = \prod_{i=1}^{n} y_i^{\,y_i}\ \ 1-y_i\ ^{(1-y_i)} \qquad \text{(I.21)}$$

Reemplazando las ecuaciones (I.20) y (I.21) en la ecuación (I.19), se tiene:

$$D = -2\sum_{i=1}^{n}\left[y_i \ln\left(\frac{\hat{p}_i}{y_i}\right) + (1-y_i)\ln\left(\frac{1-\hat{p}_i}{1-y_i}\right)\right] \sim \chi^2_{n-(k+1)} \qquad \text{(I.22)}$$

Criterio de Decisión:

No se rechaza H_0, si $D < a$

Entonces el modelo ajustado es significativo, es decir no existe diferencia entre los valores observados y los valores estimados.

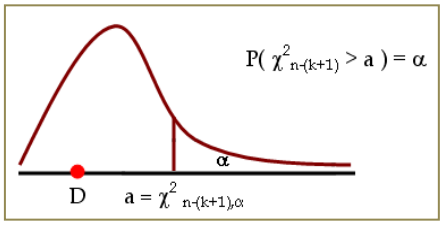


Figura I.23. *Representación gráfica de la región de rechazo de la evaluación del ajuste del modelo*

Test de Hosmer y Lemeshow:

La prueba de Hosmer-Lemeshow evalúa un aspecto de la validez del modelo: la calibración (grado en que la probabilidad predicha coincide con la observada). Para evaluar la bondad de ajuste del modelo se plantea las siguientes hipótesis considerando un nivel α de significación:

H_0 : *No existe diferencia entre los valores observados y los valores estimados a partir del modelo de regresión logística.*

H_1 : *Existe diferencia entre los valores observados y los valores estimados a partir del modelo de regresión logística.*

Estadístico de Prueba:

La prueba de Hosmer - Lemeshow agrupa los sujetos en patrones según criterios estadísticos. Primero se calculan los 9 deciles (D_1,..,D_9) de las probabilidades esperadas o estimadas $\hat{p}_i; i = 1,2,...n$ y se dividen los datos observados en 10 categorías dadas por:

$$A_j = \hat{p}_i \in \left[D_{j-1}, D_j \right) / i \in 1,2,...,n \quad ; j = 1,2,...,10$$

donde $D_0 = 0$, $D_{10} = 1$.

Se diseña una tabla de contingencia de 10 x 2, basada en las frecuencias observadas y esperadas se construye el estadístico Chi – Cuadrado de Pearson con distribución χ^2 de 8 grados de libertad.

$$T = \sum_{j=1}^{10} \frac{o_j - n_j \bar{p}_j}{n_j \bar{p}_j \left(1 - \bar{p}_j\right)}^2 \sim \chi_8^2 \qquad \text{(I.24)}$$

Donde: n_j = Número de casos en A_j ; j =1,...,10

$$o_j = \sum_{i \in A_j} y_i \quad ; j = 1,2,....,10 \; ; \quad \bar{p}_j = \sum_{i \in A_j} \frac{m_i \hat{p}_i}{n_j} \quad ; j = 1,2,....,10$$

Criterio de Decisión:

Se rechaza H_0, si $T > a$

En esta prueba se espera la ausencia de significación (T < a) lo cual indica un buen ajuste del modelo. [4]

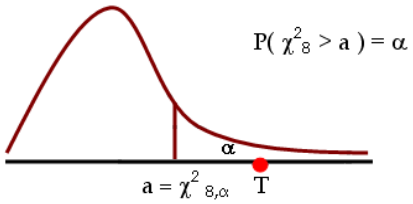


Figura I.25. *Representación gráfica de la región de rechazo del Test de Hosmer Lemeshow*

1.5.3 Prueba de Hipótesis sobre la significancia de los coeficientes

Se plantea las siguientes hipótesis:

$$H_0 : \beta_j = 0$$

$$H_1 : \beta_j \neq 0$$

Con un nivel α de significación.

Estadísticos de prueba:

$$Z = \frac{\hat{\beta}_j}{\hat{EE}(\hat{\beta}_j)} \sim N(0,1) \quad \text{ó} \quad w = \frac{\hat{\beta}_j^2}{\widehat{var}(\hat{\beta}_j)} \sim \chi_1^2$$

Criterio de Decisión:

Se rechaza H_0, si: $|Z| > Z_{\alpha/2}$ ó $w > \chi^2_{1,\alpha}$

El estadístico W es denominado Estadístico de Wald. [3]

1.5.4 Estimación por intervalos de los coeficientes

Para ello se utilizará el estadístico de prueba de la significación de los coeficientes estimados:

$$Z = \frac{\hat{\beta}_j - \beta_j}{\hat{EE}(\hat{\beta}_j)} \sim N(0,1)$$

Empleando el método pivotal se obtiene el intervalo de confianza al (1- α) % para el coeficiente β_j es:

$$IC\ \beta_j : \left\langle \hat{\beta}_j \pm Z_{\alpha/2}\, \hat{EE}(\hat{\beta}_j) \right\rangle$$

1.6 Diagnóstico del Modelo

Se realiza un diagnóstico al modelo con el fin de detectar observaciones que pueden afectar en la estimación del modelo, estas observaciones pueden ser atípicas o influyentes (outliers).

*1.6.1 **Residuos del Modelo:** Los residuos más utilizados son los siguientes:*

a. Residuos de Pearson $r_i = \dfrac{(y_i - \hat{p}_i)}{\sqrt{\hat{p}_i(1-\hat{p}_i)}} \quad i = 1,2,\ldots,n$

b. Residuos estandarizados $r^{est}{}_i = \dfrac{r_i}{\sqrt{(1-h_{ii})}} \quad i = 1,2,\ldots,n$

c. Residuos estudentizados: $r_i^* = \dfrac{(y_i - \hat{p}_{(i)})}{\sqrt{\hat{p}_{(i)}(1-\hat{p}_{(i)})}} \quad i = 1,2,\ldots,n$

donde $\hat{p}_{(i)}$ es la estimación de p_i , obtenida eliminando la i-ésima observación de la muestra.

d. Residuos desviación

$$d_i = \begin{cases} \sqrt{-2\ln(\hat{p}_i)} & Si \quad y_i = 1 \\ \\ \sqrt{-2\ln(1-\hat{p}_i)} & Si \quad y_i = 0 \end{cases} \quad ; \quad j = 1,2,\ldots,n$$

Estos residuos se distribuyen aproximadamente como una Normal Estándar ($N(0,1)$), si el modelo ajustado es correcto.

Los residuos en este caso no permiten identificar con claridad la presencia de datos discordantes, pero si son de gran ayuda para evaluar algunos supuestos del modelo, como: linealidad y homocedasticidad. [3]

1.6.2 Medidas de Influencia

Cuantifican la influencia que cada observación ejerce sobre la estimación del vector de parámetros o sobre las predicciones hechas a partir del mismo de forma que, cuanto más grande son, mayor es la influencia que ejerce una observación en la estimación del modelo.

a. Medida de Apalancamiento (Leverage): Se utiliza para detectar observaciones que pueden influir en los valores predichos por el modelo. Se calcula a partir de la matriz:

$$H = V^{1/2} X (X'VX)^{-1} X' V^{1/2}$$

El apalancamiento de la i-ésima observación viene dado por el elemento i-ésimo de la diagonal principal de H, h_{ii}, y toma valores entre 0 y 1 con un valor medio de $\frac{k}{n}$. El gráfico de h_{ii} contra $\hat{p}_i$ es de gran utilidad para detectar los posibles valores ***leverage***, es decir aquellos valores que superan al valor 2*k / n, donde k es el número de covariables y n el tamaño de la muestra. [4]

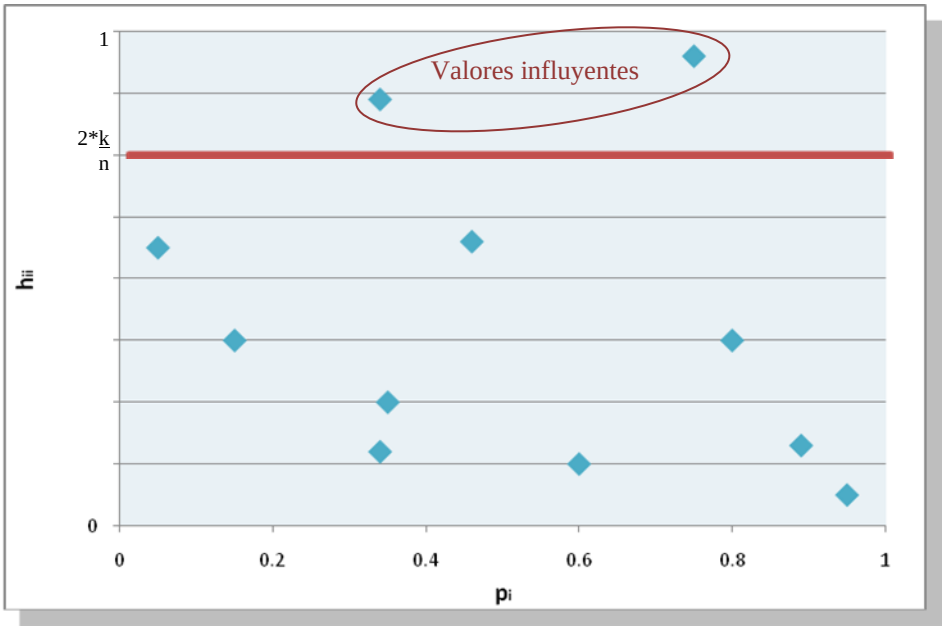


Figura I.26. ***Gráfica h_{ii} vs p_i***

b. Distancia de Cook: mide la influencia en la estimación de β. Aquellas observaciones evaluadas en la distancia de Cook y cuyo valor supere a 1 se consideran influyentes.

$$DC_i = \left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(i)}\right)' \mathbf{X}'\mathbf{V}\mathbf{X} \left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(i)}\right) = \frac{r_i^2 h_{ii}}{1 - h_{ii}^{\ 2}} \qquad i = 1, 2, ..., n$$

El gráfico de DC_i contra $\hat{p}_i$ es de gran utilidad para detectar las posibles observaciones influyentes. [3]

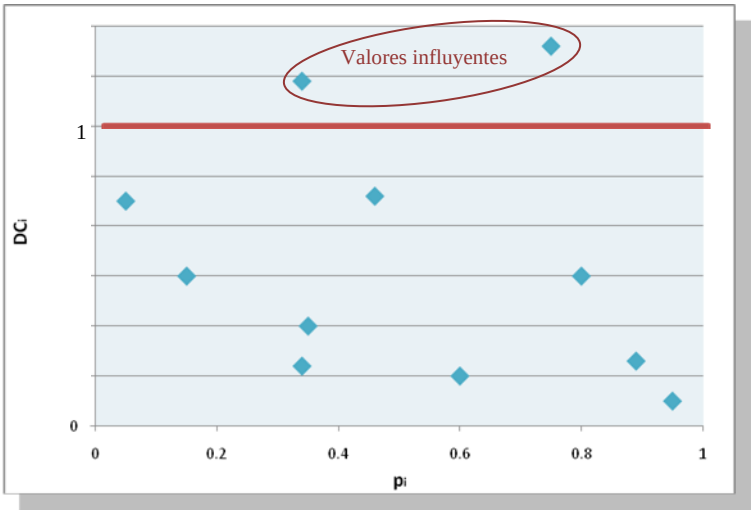


Figura I.27. ***Gráfica DC_i vs p_i***

c. Estadística DFBetas: permite detectar las observaciones que influyen en las estimaciones de cada uno de los parámetros del modelo. Permite evaluar los cambios en los parámetros estimados cuando se elimina una observación. Si el valor no se encuentra dentro de ± 2 se consideran influyentes.

$$DF\beta_{j(i)} = \frac{\hat{\beta}_j - \hat{\beta}_{j(i)}}{S(\hat{\beta}_j)} = \frac{r_i^{est} h_{ii}}{(1-h_{ii})^2} \qquad j = 0,1,....,k \quad i = 1,2,.....,n$$

donde $\hat{\beta}_{j(i)}$ denotan las estimaciones de β_j, eliminando la i-ésima observación de la muestra y $EE(\hat{\beta}_j)$ el error estándar en la estimación de $\hat{\beta}_j$. [3]

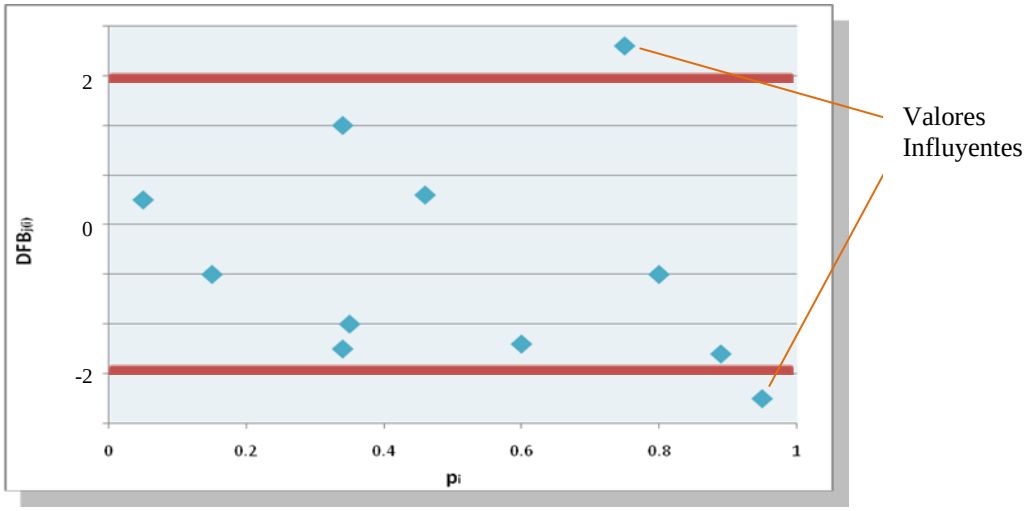


Figura I.28. ***Gráfica $DF\beta_{j(i)}$ vs p_i***

d. Estadística $\Delta\chi_i^2$: las observaciones que presentan valores grandes en $\Delta\chi_i^2$, indica que están pobremente ajustadas en el modelo. Estas observaciones tienen una gran influencia en los valores de los parámetros estimados.

$$\Delta\chi_i^2 = \frac{r_i^2}{(1-h_{ii})}$$

Toda observación para la cual $\Delta\chi_i^2$ > 4, se le puede considerar influyente. [3]

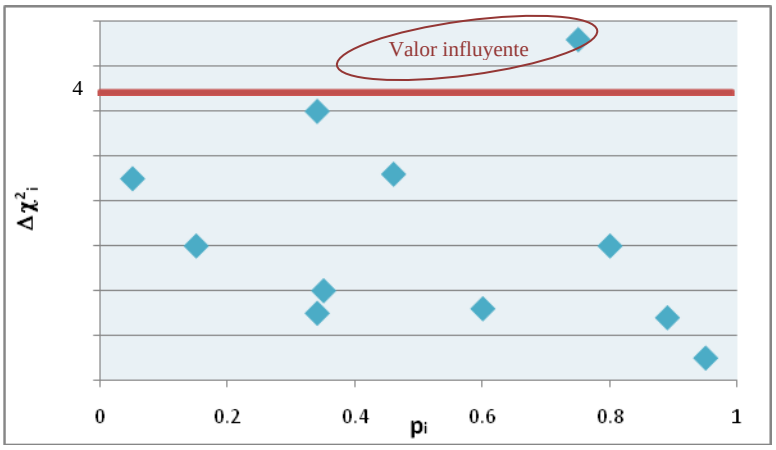


Figura I.29. ***Gráfica $\Delta\chi^2_i$ vs p_i***

e. Estadística ΔD_i: las observaciones que presentan valores grandes en ΔD_i, indica que están pobremente ajustadas en el modelo. Estas observaciones tienen una gran influencia en los valores de los parámetros estimados.

$$\Delta D_i = \frac{d_i^2}{(1-h_{ii})}$$

Toda observación para la cual ΔD_i > 4, se le puede considerar influyente. [3]

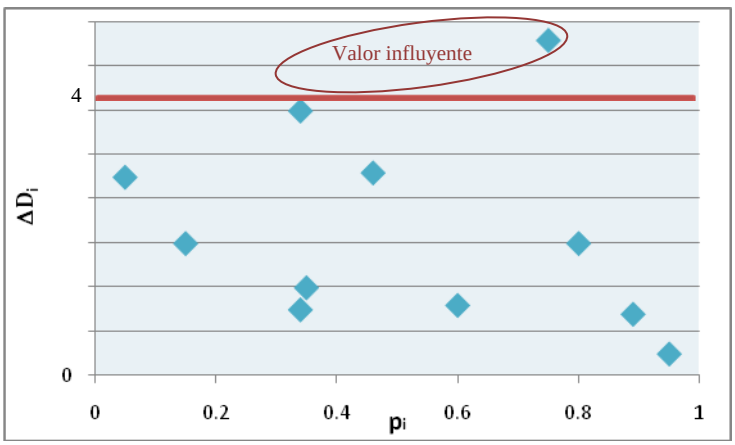


Figura I.30. ***Gráfica ΔD_i vs p_i***

1.7 Selección de Variables

Los algoritmos más utilizados para la selección de variables son:

- ***Paso a paso hacia delante (forward)***: sigue los siguientes pasos:

  a. El algoritmo se inicia incluyendo en el modelo la variable con mayor significación, contrasta el modelo con la variable frente al modelo sólo con la constante y la mantiene si la prueba de razón de verosimilitud es significativa.

  b. A continuación vuelve a evaluar cada una de las variables restantes e incorpora aquélla con mayor significación, contrasta la significación del modelo mediante la prueba de razón de verosimilitud y la mantiene si la misma alcanza significación.

  c. Se repite el paso anterior una y otra vez hasta que no quedan variables que incluir por no aportar significación.

- ***Paso a paso hacia atrás (backward):*** El algoritmo comienza incluyendo en el modelo todas las variables y elimina en cada paso aquella variable que menos contribuye a la significación del modelo, hasta mantener a todas las variables que aportan significación al modelo.

- ***Paso a paso (stepwise):*** Es el algoritmo más empleado y consiste en una combinación de los algoritmos previos. Se establecen unos criterios de entrada de variables y de salida de variables.

  a. El algoritmo se inicia con el modelo incluyendo únicamente la constante y se contrasta con los modelos surgidos de la inclusión de una única variable, e incluyendo en el modelo aquella variable que cumpliendo el criterio de entrada, tenga mayor significación.

  b. A continuación se comprueba la significación de las variables restantes una vez incorporada la primera, incluyéndose en el modelo aquélla que cumpliendo el criterio de entrada, tenga mayor significación.

c. Una vez incluida ésta, se comprueba si alcanza significación el contraste del modelo con todas las variables incluidas hasta el momento frente al modelo eliminando las incluidas en pasos previos; si alguna no alcanzara significación y alcanza el criterio de salida, dicha variable se eliminaría del modelo. Si ninguna de las variables cumplieran el criterio de salida se volvería al segundo paso.

d. El proceso terminaría cuando ninguna de las variables no incluidas en el modelo cumplieran el criterio de entrada y ninguna de las variables incluidas en el modelo cumplieran el criterio de salida. Si existieran variables clínicamente importantes que el investigador desease tener en cuanta con independencia de su significación inicial, éstas podrían ser incluidas en el modelo inicialmente junto con la constante como modelo de partida.

- ***Selección por mejores subconjuntos (best subsets):*** Consiste en el diseño y contraste de los mejores modelos con una, dos, tres… variables, y se contrastan frente al modelo con todas las variables. El mejor criterio para seleccionar "***el mejor***" de los modelos se basa en una aproximación del estadístico **C_p de Mallow** aplicado a la regresión logística, que no es ofrecido por todos los paquetes estadísticos.

Matemáticamente, la verosimilitud aumenta conforme aumenta el número de variables del modelo, cosa factible si el tamaño muestral se incrementa a la par. Sin embargo los modelos mejor interpretables son los más simples. Por ello hay estadísticos que nos permiten comparar modelos penalizando aquéllos con un mayor número de variables:

- *Criterio de información de Akaike (AIC):* -2(ln(verosimilitud) - nº parámetros)
- *Criterio de información bayesiano (BIC): G - grados de libertad · ln(tamaño muestral)*

Según estos estadísticos, son mejores los modelos con menores AIC y BIC. [4]

1.8 Interpretación

A diferencia del modelo de regresión múltiple donde el valor de un coeficiente significa el cambio en unidades de la variable dependiente por cada unidad de la variable independiente; en el caso del modelo de regresión logística la interpretación se realiza en función del coeficiente estimado exponencialmente.

Por ello se calculará el coeficiente exponencial y luego se procederá a estimar. Para ello se divide la *ecuación (I.4)* por $(1-p)$, esto es, por la probabilidad de que ocurra el otro resultado. Teniendo en cuenta que $(1-p)$ es:

$$1-p=1-\frac{e^{X'\beta}}{1+e^{X'\beta}}=\frac{1}{1+e^{X'\beta}} \qquad \text{I.31}$$

Obteniendo el cociente de probabilidades conocido como ***ODDS RATIO***:

$$\text{OR} = \frac{p}{1-p}=e^{X'\beta} \qquad \text{I.32}$$

En donde:

OR = 1, no indica nada no influye la variable en el riesgo.

OR < 1, indica que la variable es un factor protector.

OR > 1, indica que la variable es un factor de riesgo.

Estimando el ***ODDS RATIO*** se tiene:

$$\hat{OR}=\frac{\hat{p}}{1-\hat{p}}=e^{X'\hat{\beta}} \qquad \text{I.33}$$

Se debe tener en cuenta que los estimadores de la asociación no son los coeficientes β_i sino los ODDS RATIO, por lo tanto los intervalos de confianza que interesan calcular son de los ODDS RATIO. Cuyos intervalos están dados por:

$$e^{\left[\hat{\beta}_i \pm Z_{\alpha/2} EE(\hat{\beta}_i)\right]} \qquad \text{I.34}$$

1.9 Ventajas

- El análisis de regresión logística es una herramienta muy flexible en cuanto a la naturaleza de las variables explicativas, pues éstas pueden ser de escala y categóricas; a diferencia del análisis de regresión lineal que sólo trabaja con variables independientes de escala métrica. [5]

- El modelo de regresión logística es robusto con respecto al incumplimiento del supuesto de igualdad de las matrices de covarianza entre grupos (heterocedasticidad).

- La regresión logística no hace supuestos sobre la distribución de las variables independientes, ya que en el modelo estas no consideradas como variables aleatorias.

- Tamaño de muestra y número de variables independientes. Una de las ventajas de la regresión logística es que permite el uso de múltiples variables con relativamente pocos casos, sin embargo, hay que tener en cuenta algunas precauciones. Para poder usar esta técnica estadística sin problemas el número de sujetos debe ser superior a 10(k+1) donde k es el número de variables explicativas; por tanto, si se introducen interacciones o variables *dummy*, el número de elementos en la muestra debe aumentar. Además se ha sugerido que si una de las variables dicotómicas (en especial si es la de respuesta) no tiene al menos 10 casos en cada uno de sus 2 valores posibles, entonces las estimaciones no son confiables. En cuanto al número de variables independientes, la inclusión de un gran número de ellas en el modelo (ej. K>15), puede indicar que no se ha reflexionado suficientemente sobre el problema. [6]

- Es necesario tener en cuenta el efecto sobre el riesgo de que ocurra el evento, de los cambios de las variables explicativas cuando son cuantitativas (continuas), en ocasiones es necesario categorizarlas, ya que los cambios que se producen de una unidad a otra pueden resultar intrascendentes o no ser constantes a lo largo del rango de valores de la variable. [6]

- Cuando algunas de las variables independientes analizadas están altamente correlacionadas, los resultados que se obtienen pueden no ser satisfactorios, por esta razón debe realizarse un análisis univariado previo entre las distintas variables explicativas. [6]

1.10 Desventajas

- Modelos excesivamente grandes para muestras con tamaños muestrales pequeños implicaran errores estándar grandes o coeficientes estimados elevados (sobreajuste). [4]

- Dado que el modelo de regresión logística no es lineal, la estimación de sus parámetros será compleja, es decir, por medio de procedimientos recursivos se podrá encontrar el verdadero valor de los mismos. Uno de los métodos que se utilizan es el de **Newton-Raphson**. [7]

- La multicolinealidad entre las covariables, traerá como consecuencia grandes errores estándar y coeficientes estimados anormalmente elevados. [4]

CAPÍTULO II:

Árboles de Clasificación

CART

2.1 Introducción

El análisis del árbol de clasificación llamado también de decisión o de identificación, es una técnica de segmentación diseñada para dividir a una población en dos o más grupos basándose en sus atributos, por ejemplo: se desea conocer que segmento de personas se encuentran predispuestas a tener hipertensión (género, edad, antecedentes familiares, etc.); en general este método es utilizado en diagnóstico médico, análisis de riesgo en la concesión de créditos y elaboración de horarios.

El modelo de árbol de clasificación presentan un estructura en forma de árbol, en donde las ramas representan conjuntos de decisiones; estas decisiones generan sucesivas reglas para la clasificación de un conjunto de datos en subgrupos de datos disjuntos y exhaustivos. Las ramificaciones se generan de forma recursiva hasta que se cumplan ciertos criterios de parada. **[8]**

La construcción automática de árboles de decisión parte de los trabajos en ciencias sociales de Morgan y Sonquist (1963) y Morgan y Messenger (1973). Breiman (1984) influyo tanto en despertar el interés de los estadísticos

como en el de proponer nuevos algoritmos para la construcción de árboles. Sobre esa misma época la inducción mediante árboles de decisión comenzó a usarse en el ***"Aprendizaje de las Máquinas"*** (Inteligencia Artificial), especialmente por Quinlan (1979, 1983, 1986, 1993). Sin embargo no hubo una amplia literatura al respecto, estando dispersas la mayoría de las contribuciones estadísticas. Clark y Pregibon (1992) describen los modelos basados en árboles y los implementan en lenguaje S, haciendo que estos métodos sean mucho más asequibles. Sus métodos son muy flexibles y están fundamentalmente orientados al análisis exploratorio de datos. [9]

Los *árboles de decisión* son empleados para *clasificar* y *pronosticar*, es decir identificar el resultado categórico atendiendo a una serie de criterios dados y pronosticar el *resultado* según una futura serie de criterios o *variables independientes*.

El objetivo de este método es obtener individuos u objetos más homogéneos con respecto a la variable discriminadora dentro de cada subgrupo y heterogéneos entre los subgrupos. Para la construcción del árbol se requiere información de variables explicativas a partir de las cuales se va a realizar la discriminación de la población en subgrupos.

Dentro de los métodos basados en árboles se pueden distinguir dos tipos dependiendo de tipo de variable a discriminar:

- ***Árboles de clasificación.*** Este tipo de árboles se emplea para variables categóricas, tanto nominales como ordinales.
- ***Árboles de regresión.*** Este tipo de discriminación se aplica a variables continuas.

2.2 Definición

Los análisis de clasificación basados en árboles de decisión son técnicas de explotación de datos (*Data Mining*) que consisten en estudiar grandes masas de datos con el fin de descubrir patrones.

Un árbol de decisión tiene como entrada la descripción de un conjunto de atributos $X = X_1, X_2, ..., X_k$ y mediante estos devuelve una respuesta Y , la cual viene a ser una decisión tomada a partir de las entradas. Los valores que pueden tomar las entradas y las salidas pueden ser valores discretos o continuos. Se utilizan más los valores discretos por simplicidad, cuando se utilizan valores discretos en las funciones de una aplicación se denomina clasificación y cuando se utilizan los continuos se denomina regresión.

Un árbol de decisión lleva a cabo un test a medida que este se recorre hacia las hojas para alcanzar así una decisión. El árbol de decisión suele contener nodos internos, nodos de probabilidad, nodos hojas y arcos. Un nodo interno contiene un test sobre algún valor de una de las propiedades. Un nodo de probabilidad indica que debe ocurrir un evento aleatorio de acuerdo a la naturaleza del problema, este tipo de nodos es redondo, los demás son cuadrados. Un nodo hoja representa el valor que devolverá el árbol de decisión. Y finalmente las ramas brindan los posibles caminos que se tienen de acuerdo a la decisión tomada. [10]

Si llamamos Ω al espacio muestral de la variable que deseamos clasificar, es decir, a su conjunto de clases, el objetivo de un árbol es conseguir una partición de dicho espacio. Dos árboles se pueden comparar en el sentido de que su partición inducida se aproxime en mayor o menor medida a la regla de decisión correcta para el problema. En problemas de tipo lógico el modo más fácil de comparar particiones es contar el número de errores que cometemos en la clasificación, o bien, si disponemos de una partición a priori sobre Ω, calculando la probabilidad de error.

Algoritmos

Los árboles de decisión generalmente son construidos con la ayuda de un algoritmo, el cual divide los registros en grupos; la probabilidad del resultado es diferente en cada grupo atendiendo a los valores de las variables independientes. Existe una gran variedad de algoritmos de árboles de decisión: [11]

- ***CHAID.*** – es un algoritmo estadístico rápido y multidireccional que explora rápida y eficientemente datos, y construye segmentos y perfiles en función de la variable de respuesta establecida. Fue introducido por Kass en 1980, y es un derivado del THAID [12]. El criterio para particionar está basado en χ^2 y para terminar el proceso se requiere definir de antemano un "*threshold*" (umbral).

- ***CHAID Exhaustivo.*** – es una modificación del CHAID que examina todas las posibles particiones de la variable predictora.

- ***Árboles de Clasificación y Regresión (C&RT ó CART).*** – es un algoritmo binario completo que hace particiones de datos y produce subconjuntos homogéneos precisos. Fue diseñado por L. Brieman. Existe una versión similar llamada IndCART distribuido por la NASA. El criterio para particionar es la impureza del nodo.

- ***QUEST.*** – es un algoritmo estadístico que selecciona variables de manera no sesgada y construye árboles binarios precisos rápida y eficientemente.

- ***ID3.*** – es un algoritmo que genera árboles de decisión a partir de ejemplos de partida.

- ***C4.5.*** – es un algoritmo basado en ID3 (Interactive Dichotomiser 3) de J. Ross Quinlan (1983) dentro de la comunidad de "Machine Learning".

- ***NewId.*** – es un algoritmo muy similar a C4.5.

- ***Árboles Bayesianos.*** – es un algoritmo basado en la aplicación de métodos Bayesianos a árboles de decisión. Buntine (1992).

- ***CN2 -*** Introducido por Clark and Niblett (1988).

2.3 Estructura de un Árbol de Clasificación

Partiendo de una Base de Datos con una variable ***Y*** a discriminar, denominada variable respuesta, y un conjunto finito de variables $X_1, X_2,..., X_k$ conocidas como variables explicativas. Se tratará de seleccionar entre las variables explicativas aquellas que discriminen mejor a la variable ***Y***. Obteniéndose una partición de la población de forma que se encuentren dos o más subgrupos lo más heterogéneos posibles entre sí con respecto a la variable respuesta ***Y***, y lo más homogéneos posibles dentro. Esta discriminación se continúa para los nuevos nodos generados y se aplica un criterio de parada, obteniendo el árbol de clasificación o regresión.

Todo árbol de clasificación comienza con un nodo al que pertenecen todos los casos de la muestra a clasificar (*nodo raíz*), el resto de nodos se dividen en *nodos intermedios* o no terminales y *nodos hojas* o nodos terminales.

Un árbol de decisión consta de los siguientes elementos:

- ***Nodos intermedios:*** se generan dos o más segmentos descendientes inmediatos (dependiendo del método empleado). También llamados segmentos intermedios.
- ***Nodos terminales:*** Es un nodo que no se puede dividir más. También denominado segmento terminal.
- ***Rama de un nodo t :*** Consta de todos los segmentos descendientes de t, excluyendo t .
- ***Árbol de decisión completo (A_{max}):*** Árbol en el cual cada nodo terminal no se puede ramificar.
- ***Sub-árbol:*** Se obtiene de la poda de una o más ramas del árbol A_{max}.

A pesar de los distintos tipos de árboles de clasificación y regresión existentes la forma de actuar en todos ellos es similar, salvo ligeras modificaciones. En primer lugar se debe tener un conjunto de datos con una variable respuesta (categórica o continua) y un conjunto de variables explicativas, todas ellas categóricas o continuas que han sido previamente categorizadas. Todos los registros de la base de datos son examinados para encontrar la mejor regla de clasificación de la variable respuesta. Estas reglas

se realizan basándose en los valores de las variables explicativas. La secuencia de particiones define el árbol. Cada partición se realiza para optimizar la clasificación del subconjunto de datos. El proceso de división es recursivo y finaliza la ramificación cuando se verifica un criterio de parada que ha debido ser definido previamente. [13]

Por lo tanto, la construcción de un árbol de decisión se basa pues en cuatro elementos:

a) Un conjunto de preguntas binarias Q de la forma { x ∈ A?}, donde A es un subconjunto del espacio muestral.

b) El método usado para particionar los nodos.

c) La estrategia requerida para parar el crecimiento del árbol.

d) La asignación de cada nodo terminal a un valor de la variable de respuesta (regresión) o a una clase (clasificación).

Las diferencias principales entre los algoritmos para construir árboles se hallan en la regla para particionar los nodos, la estrategia para podar los árboles, y el tratamiento de valores perdidos. [11]

2.3.1 Formación de Nodos

Hay un gran número de posibles formas de efectuar divisiones en función de los valores que tomen las variables explicativas X_1, X_2,..., X_k, y generalmente no se pueden considerar todas ellas. Dependerá en gran medida del tipo de variable que estemos tratando: [13]

- **Variable cualitativa nominal*:*** En este caso la variable toma *C* valores distintos entre los que no cabe establecer un orden natural. Si tenemos que discriminar con ayuda de una variable nominal los elementos que van a los distintos nodos hijos en el nodo t, podemos formar todos los subgrupos de los *C* valores que puede tomar X_i y enviar a un nodo los casos que generan la mejor discriminación con respecto a la variable respuesta y los restantes al otro nodo.

- **Variable cualitativa ordinal:** En este caso si la variable toma *d* valores, una vez ordenadas las categorías, se consideran como posibles cortes los *d-1* valores intermedios. Entre estos posibles cortes se considerará el que proporcione grupos más homogéneos con respecto a la variable respuesta.

- **Variable cuantitativa continua:** Se trabaja con estas variables de la misma forma que con las variables ordinales, con la particularidad de que en este caso el número de valores de corte a comprobar será elevado debido al caso de no haber repeticiones, *n-1* cortes en el caso de ser *n* el tamaño de la muestra. De este conjunto se seleccionarán los grupos que mejor discriminen los individuos con respecto a la variable respuesta.

2.4 Medidas de la Impureza

2.4.1 Impureza de un nodo:

Para decidir qué variable va a utilizarse para hacer la partición en un nodo se calcula primero la proporción de observaciones que pasan por el nodo para cada uno de los grupos. Si se denomina a los nodos como $t = 1, 2, ..., T$ y $p\ \ g|t$ a las probabilidades de que las observaciones que lleguen al nodo t pertenezcan a cada una de las clases [14], se define la impureza del nodo t como:

$$i(t) = \phi\ \ p\ \ 1|t\ \ , p\ \ 2|t\ \ , ..., p\ \ G|t \qquad \text{(II.1)}$$

Donde: ϕ es la función de impureza y, p (g| t) puede calcularse empíricamente como la proporción de casos de clase *g* en el nodo *t*: [15] Es decir:

$$p\ \ g|t\ \ = \frac{n_g(t)}{n(t)} \qquad \text{(II.2)}$$

La variable que se introduce en un nodo es la que minimiza la heterogeneidad o impureza que resulta de la división en el nodo. La clasificación de las observaciones en los nodos terminales se hace asignando todas las observaciones del nodo al grupo más probable en ese nodo, es decir, el grupo con máxima $p\ g|t$. Si la impureza del nodo es cero, todas las observaciones pertenecerían al mismo nodo, en caso contrario puede haber cierto error de clasificación. Cuando el número de variables es grande, el árbol puede contener un número excesivo de nodos por lo que se hace necesario definir procedimientos de poda o simplificación del mismo. [14]

2.4.2 Bondad de una partición

La bondad de una partición *s* en un nodo *t* debe estar relacionada con la impureza del nodo sobre el que se realiza la partición *t*, y con la impureza de los nodos resultantes de la partición, t_L y t_R. Supongamos una partición *s*, que divide *t* en t_L y t_R de forma que una proporción p_L de los casos de *t* van a t_L y una proporción p_R van a t_R.

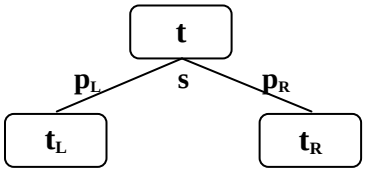


Figura II.3. ***Ejemplo de una partición***

La bondad de la partición s en un nodo t, ϕ(s, t) se define como el decrecimiento en impureza conseguido con ella:

$$\phi(s,t) = \Delta i(s,t) = i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R) \qquad \text{(II.4)}$$

Así, como conocemos cómo calcular *i(t)*, podemos calcular $\phi(s,t)$ para cada partición s y seleccionar la mejor partición como la que proporciona la mayor bondad $\phi(s,t)$. Para establecer el efecto que produce la selección de la mejor partición en cada nodo sobre el árbol final necesitamos una medida de la impureza global del árbol. [15]

2.4.3 Impureza de un árbol

La ***impureza del árbol T***, denotado por *I*(*T*), se define como:

$$I(T) = \sum_{t \in T} I(t) = \sum_{t \in T} p\ t\ \cdot i(t) \qquad \text{(II.5)}$$

donde *p*(*t*) es la probabilidad de que un caso cualquiera esté en el nodo *t*.

La impureza de un árbol se calcula únicamente en base al conjunto de nodos terminales. Además la selección continuada de las particiones que maximizan Δi(s, t) es equivalente a seleccionar las particiones que minimizan la impureza global *I(T)*, lo que significa que la estrategia de selección de la mejor partición en cada nodo conduce a la solución óptima considerando el árbol final. [15]

2.4.4 Estimación de la tasa de error

La elección de un árbol respecto de otro dependerá en general de una estimación de su tasa de error R(T). El problema es cómo realizar la estimación de dicha tasa, por ello existen diversas formas de calcular la estimación con una serie de ventajas e inconvenientes que se detallan a continuación: [13]

- **Estimador por resustitución (estimación intramuestral):** Es el estimador más simple. Consiste en dejar caer por el árbol la misma muestra que ha servido para construirlo, pero como los árboles tienen gran flexibilidad para adaptarse a la muestra se puede obtener una estimación sesgada inferiormente de la tasa de error, y por tanto desconocer realmente el error real del árbol.

- **Estimador por muestra de validación (muestra de contraste):** Consiste en dejar caer por el árbol una muestra distinta a la empleada para la realización del árbol. Por ello éste no se ha podido adaptar a dichos registros como ocurría en el estimador anterior. Tenemos de esta forma un estimador de R(T) insesgado, sin embargo este tiene el inconveniente de forzar a reservar, para su uso en la validación, una parte de la muestra la cual podía haberse empleado en la construcción del árbol. Por lo que hay cierta pérdida de información. Este estimador es empleado cuando se tiene tamaño de muestra muy grande, como en el caso de los

censos, debido a que no se pierde mucha información al eliminar del estudio una muestra para la estimación del error.

- **Estimación por validación cruzada:** consiste en estimar R(T) procediendo de forma reiterada similar al estimador por muestra de validación. Se deja fuera de la muestra a una fracción m^{-1} del tamaño muestral total para la construcción del árbol. Obteniéndose de esta forma *m* estimaciones $R^{(1)}(T),.....,R^{(m)}(T)$ y promediándolas de la siguiente forma:

$$R^{VC}(T) = \frac{R^{(1)}(T) + ... + R^{(m)}(T)}{m} \qquad \text{(II.6)}$$

  Observar que el árbol realizado para cada una de las submuestras podría ser distinto a los demás, en este caso la expresión anterior no sería válido.

- **Estimador bootstrap:** Recientemente se ha propuesto esta técnica de remuestreo para la estimación de la tasa de error. Ripley (1996).

2.4.5 Reglas de parada

Existen distintos criterios de parada que pueden provocar la finalización de los algoritmos que realizan árboles de clasificación o regresión. Las causas que pueden provocar la finalización son: [13]

- Se ha alcanzado la máxima profundidad del árbol permitida.
- No se pueden realizan más particiones, porque se ha verificado alguna de las siguientes condiciones:
  - *a.* No hay variables explicativas significativas para realizar la partición del nodo.
  - *b.* El número de elementos en el nodo terminal es inferior al número mínimo de casos permitidos para poder realizar la partición.
  - *c.* El nodo no se podrá dividir en el caso en el cual el número de casos en uno o más nodos hijos sea menor que el mínimo número de casos permitidos por nodo.

2.4.6 Técnicas Básicas en la Construcción de los Árboles:

Existen dos técnicas básicas en la construcción de los árboles:

- ***Mirada hacia delante***.- Está técnica se basa en subdividir los nodos escogiendo en cada momento la división que produzca la máxima disminución de impureza *I(t)* mientras un estimador adecuado de la tasa de error *R(T)* disminuya. Dado que en cada paso se examinan árboles con un número de nodos muy similar, basta estimar *R(T)* por $R(\hat{T})$. En el momento en el que no se obtiene un descenso de la tasa de error aceptable se para la fase de la ramificación y se considera a este como el árbol óptimo.

- ***Mirada hacia atrás***.- Está técnica sugiere construir árboles frondosos, llegando al árbol máximo posible A_{max} sin tener en cuenta las tasas de error y tras su construcción se procede a realizar un trabajo de poda y quedarnos con aquel árbol que proporcione menor tasa de error. Esta teoría se basa en que no se conoce lo que hay tras una ramificación si no se realiza y en el caso de no encontrar resultados satisfactorios siempre estaremos a tiempo de eliminar dicho rama. Tras construir el árbol completo A_{max} se aplica un algoritmo de poda con el cual se obtiene una secuencia de sub-árboles mediante la supresión sucesiva de las ramas que proporcionan menos información en términos de discriminación entre las clases de la variable Y. Finalmente se elige el sub-árbol *A** que proporcione la menor tasa de error.

Una posibilidad de poda para los árboles de clasificación consiste en el uso de la tasa de mala clasificación. Esta es una medida del porcentaje de casos mal clasificados en un nodo terminal. Se crea la función indicadora $\chi \bullet$ que valdrá 1 en el caso en que la condición incluida entre los paréntesis sea cierta y 0 en caso contrario. Por tanto la tasa de mala clasificación ***R(d)*** será calculada de la siguiente forma: [13]

$$R(d) = \frac{1}{n}\sum_{i=1}^{n} \chi \ \ d(x_i) \neq g_i \qquad \text{(II.7)}$$

donde:

n: número total de casos que han sido clasificados.
$d(x_i)$: categoría asociada al nodo para el caso i.
g_i : verdadera categoría del caso i.

2.5 Modelo del Árbol de Clasificación: CART

El procedimiento para este modelo no utiliza un modelo estadístico formal, para clasificar se utilizan particiones binarias sucesivas de los valores de una variable, dichas variables pueden ser de tipo cualitativas o cuantitativas.

2.5.1 Construcción del árbol de clasificación

Supongamos que el vector de variables predictoras es X, donde algunas de las variables X_i son cualitativas y otras son cuantitativas. Entonces, el conjunto Q de preguntas binarias en los nodos debe tener las siguientes características:

a. Cada división de los nodos depende del valor de una sola variable predictora.

b. Si la variable Xi es continua entonces Q incluye todas las preguntas de la forma {Es Xi ≤ c}, donde c es cualquier número real. Usualmente c es el punto medio entre dos valores consecutivos de un atributo.

c. Si la variable Xi es categórica tomando valores en $\{b_1, b_2, \ldots\ldots, b_m\}$ entonces Q incluye todas las preguntas de la forma $\{ X_i \in A?\}$ donde A es un subconjunto cualquiera de $\{b_1, b_2, \ldots\ldots b_m\}$. En total se pueden considerar $2^{m-1}-1$.

Un árbol de decisión particiona el espacio de variables predictoras en un conjunto de hiper-rectángulos y en cada uno de ellos ajusta un modelo sencillo, generalmente una constante. Es decir y = c, donde y es la variable de respuesta.

La idea fundamental es que los nodos hijos sean más puros que los nodos padres. La partición de un nodo t del árbol T se hace de acuerdo a un criterio que es diseñado para producir nodos hijos que produzcan una suma de cuadrados de errores menor que separen mejor las clases que el del nodo padre en el caso de clasificación.

En árboles de clasificación sean $p(s) = \frac{\#\{i \le n : X_i \in s\}}{n}$ la proporción de observaciones en el nodo s, y $p(g|s) = \frac{\#\{i \le n : X_i \in s \quad y \quad Y_i = g\}}{\#\{i \le n : X_i \in s\}}$ la proporción de observaciones en el nodo **s** que pertenecen a la clase **g** (g = 1,.., G), donde G es el número de clases.

El índice de la impureza del nodo t como $i(t) = \phi\ p\ 1|t\ , p\ 2|t\ , ..., p\ G|t$ donde ϕ es una función de impureza, la cual debe satisfacer ciertas propiedades. Entonces la regla para particionar el nodo t es formar el nodo hijo derecho t_R y el nodo hijo izquierdo t_L tal que la disminución de la impureza dada por: $\phi(s,t) = \Delta i(s,t) = i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R)$ sea máxima. [11]

Para árboles de clasificación se pueden usar las siguientes medidas de impureza:

a. ***El Coeficiente de Gini****:* El índice de Gini en el nodo t se define por:

$$i\ t\ = g\ t\ = 1 - \sum_{g=1}^{G} p\ g|t\ ^2 \qquad \text{II.8}$$

Este índice es una medida de impureza en la clasificación de los datos, a medida que ve van clasificando correctamente los datos, el índice de Gini va tomando valores cercanos a 0.

b. ***La Entropía Cruzada o Devianza o Impureza de Información:*** definida por:

$$i\ t\ = e\ t\ = -\sum_{g=1}^{G} p\ g|t\ \times log\ \ p\ \ g|t \qquad \text{II.9}$$

En esta medida se asume que: ***0 x log(0) = 0***.

c. ***La tasa de Mala clasificación:*** definida por:

$$i\ t\ = MC\ t\ = 1 - max_g\ \ p\ \ g|t \qquad \text{II.10}$$

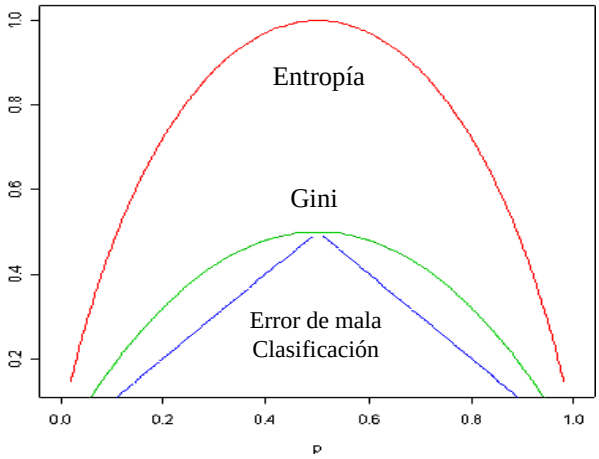


Figura II.11. ***Representación gráfica de las medidas de impureza***

2.5.2 Árboles basados en Modelos de Segmentación de Recursivos Binarios

Se define un árbol binario como un grafo formado por nodos y arcos verificando lo siguiente:

1. Hay un solo nodo que no tiene padre y se denomina raíz.
2. Cada nodo distinto de la raíz tiene un único padre.
3. Cada nodo tiene exactamente dos o ningún hijo. En el caso de nodos sin hijos o nodos terminales hablamos también de hojas.

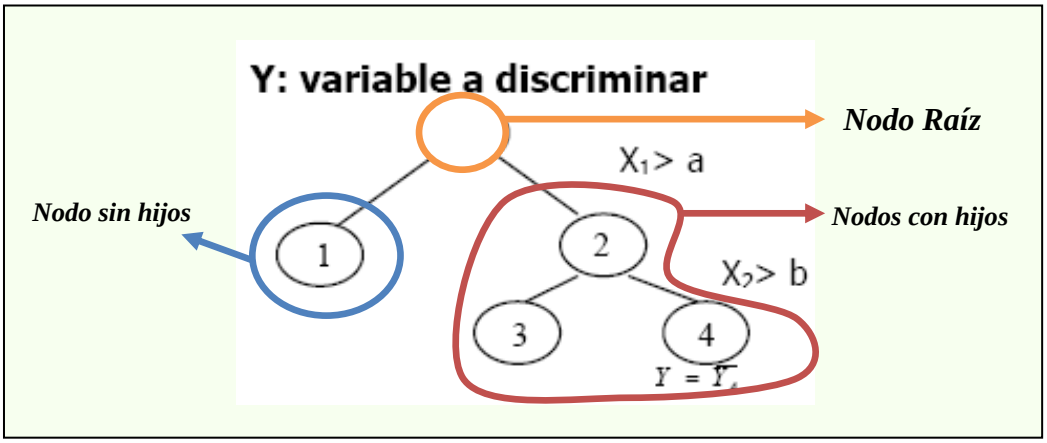


Figura II.12. ***Esquema de un árbol binario***

Podemos ver un árbol binario como una representación esquemática de un proceso de partición recursiva, en el cual en cada nodo no terminal se toma la decisión de dividir la muestra de una cierta manera.

La idea básica de la segmentación recursiva binaria consiste en ir dividiendo los datos en sucesivas particiones binarias. Tras un nodo padre se generan dos nodos hijos dividiendo los individuos pertenecientes al nodo padre en base a los valores de una variable explicativa. Se emplea para la partición la variable explicativa que mejor discrimina a la variable respuesta. El algoritmo actúa de forma recursiva y los nodos hijos generados pasan a ser potenciales nodos padres que a su vez pueden generar otro par de nodos hijos. El objetivo de las sucesivas ramificaciones y la construcción del árbol es obtener grupos de elementos homogéneos dentro de los nodos y heterogéneos entre los distintos nodos. El algoritmo procede a evaluar todos los posibles nodos padres candidatos a ramificar y selecciona aquel que más reduce la heterogeneidad dentro del nodo si se procede a generar dos hijos a partir de él. Esto se realiza sucesivamente, cuanta mayor profundidad del árbol menor será el número de individuos pertenecientes a cada nodo hasta llegar a un punto en el cual no se pueda realizar más ramificaciones y se obtengan los llamados nodos terminales.

2.6 Ventajas

Entre las ventajas de esta técnica no paramétrica de clasificación de la población están las siguientes: [13]

- Es una técnica no paramétrica que tiene en cuenta las interacciones que pueden existir entre los datos.
- Es robusta frente a datos atípicos o individuos mal etiquetados.
- Es válida sea cual sea la naturaleza de las variables explicativas: continuas, nominales u ordinales.

- Los criterios de construcción del árbol, el método y el algoritmo son los mismos tanto para árboles de clasificación como para los de regresión.

- Las reglas de asignación son legibles y por tanto la interpretación de resultados es directa e intuitiva.

2.7 Desventajas:

Las desventajas de esta técnica no paramétrica de clasificación de la población están las siguientes: [13]

- Dificultad para elegir el árbol óptimo.

- Las reglas de asignación son bastantes sensibles a pequeñas perturbaciones en los datos (inestabilidad).

- Ausencia de una función global de las variables y como consecuencia pérdida de la representación geométrica.

- Los árboles de clasificación requieren un gran número de datos para asegurarse que la cantidad de las observaciones de los nodos hoja es significativa.

CAPÍTULO III:

Redes Neuronales

3.1 Introducción

Antes de definir que son los modelos de Redes Neuronales, se debe conocer cual fue su origen y evolución en el transcurso del tiempo.

Los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Beltran Russell, Warren McCulloch y Walter Pitts, quienes en 1943 lanzaron una teoría [16] acerca de la forma de trabajar de las neuronas. Ellos modelaron una red neuronal simple mediante circuitos eléctricos. Entre los años 1940 y 1950, los científicos comenzaron a pensar seriamente en la Red Neuronal utilizando como concepto la noción de que las neuronas del cerebro funcionan como interruptores digitales (on-off) de manera similar se desarrolló el computador digital, así nace la idea de la **revolución cibernética**.

En 1956, el Congreso de Dartmouth indico el nacimiento de la inteligencia artificial. Frank Rosenblatt comenzó el desarrollo del Perceptron en 1957, este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado en el entrenamiento. Sin embargo, tenía una serie de limitaciones, por ejemplo, su incapacidad para resolver el problema de la función OR exclusiva y era incapaz de determinar clases no separables linealmente. En su libro [17] confirmó que bajo ciertas condiciones el aprendizaje

del Perceptron convergía hacia un estado finito.

Bernard Widroff y Marcian Hoff desarrollaron los modelos Adaline (**ADA**ptative **LIN**ear **E**lements) y Madaline (**M**ultiple **ADALINE**) en 1960, estos fueron aplicados a un problema real de filtros adaptativos para eliminar ecos en las líneas telefónicas. Rosemblat publicó en 1962 los resultados de las capacidades del Perceptron al desarrollar la regla de aprendizaje delta, que permitía emplear señales continuas de entrada y salida.

Una de las contribuciones más importantes realizadas en el campo de la estadística en las últimas décadas del siglo XX ha sido la introducción del concepto de modelo lineal generalizado (MLG) por parte de J. Nelder y R. W. Wedderburn. El MLG constituye la generalización natural de los modelos lineales clásicos como: regresión lineal, análisis de varianza, análisis de covarianza, regresión de Poisson, regresión logística, regresión logit, modelos log-lineales, modelos de respuesta multinomial, así como ciertos modelos de análisis de supervivencia y de series temporales.

Paul Werbos desarrolló la idea básica del algoritmo de aprendizaje de *propagación hacia atrás* (backpropagation) en 1974; cuyo significado quedó definitivamente aclarado en 1985.

A partir de 1986, el panorama fue alentador con respecto a las investigaciones y el desarrollo de las redes neuronales. En 1988 fue formada la Sociedad Internacional de Redes Neuronales. En la actualidad, son numerosos los trabajos que se realizan y publican cada año, las aplicaciones nuevas que surgen (sobretodo en el área de control) y las empresas que lanzan al mercado productos nuevos, tanto hardware como software (sobre todo para simulación).

3.2 Redes Neuronales (RNA).

Las RNA son definidas de muchas formas, como:

- *"Una nueva forma de computación, inspirada en modelos biológicos"*. [18]
- *"Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles".* [18]
- *"Un sistema de computación compuesto por un gran número de*

elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas". [18]

- *"Redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico".* [18]

- *"Un sistema compuesto de muchos elementos simples de proceso operando en paralelo cuya función está determinada por la estructura de la red, los pesos de las conexiones, y el procesado realizado en los elementos o nodos de cálculo".* [19]

- *"Un sistema procesador de información con características de desempeño similares a las redes neuronales biológicas. Una red neuronal artificial ha sido desarrollada como la generalización de modelos matemáticos del conocimiento humano o biología neuronal, basada en las siguientes acepciones: El procesamiento de información ocurre en elementos sencillos llamados neuronas. Las neuronas se transmiten señales a través de ligas de conexión. Cada liga de conexión esta asociada con un peso, el cual, en típicas redes neuronales, multiplica la señal transmitida. Cada neurona aplica una función de activación (usualmente no lineal) a la entrada de la red (la suma de entradas ponderadas por los pesos)".* [20]

- *"Un procesador distribuido y con estructura paralela que tiene una tendencia natural a almacenar conocimiento experimental, haciéndolo apto para su uso. Se parece al cerebro en dos cosas:*
  - *El conocimiento es adquirido por la red a través de un proceso de aprendizaje.*
  - *El conocimiento almacenado en los pesos sinápticos o conexiones entre neuronas."* [21]

- *"Un circuito compuesto de un número elevado de elementos simples de proceso con una base neurológica. Cada elemento opera sólo con información local. Más aún, cada elemento opera asíncronamente por lo*

que no hay un reloj total del sistema". [22]

3.2.1 Definición de la Red Neuronal

Las Redes Neuronales o Redes Neuronales Artificiales (RNA) son técnicas de explotación de datos, es decir extrae datos de una información implícita, desconocida y potencialmente útil. Las Redes Neuronales comprenden muchos modelos y métodos de aprendizaje.

Una red neuronal consiste en un modelo de nodos e interconexiones que recrea el diseño de las interconexiones neuronales del cerebro humano. Respecto al modo interno de trabajo las redes neuronales son modelos matemáticos multivariantes que utilizan procedimientos iterativos, con el objetivo de minimizar una determinada función de error.

Con las redes neuronales se puede realizar automática y eficientemente múltiples tareas como: modelación, optimización, ***regresión, clasificación***, lógica difusa, patrones y rasgos ocultos, memorización, aprendizaje asociativo, control adaptativo, ***Pronóstico y Predicción de Series de Tiempo***, etc.

Los diferentes tipos de RNA son sistemas definidos por funciones denotados por *f(·)*. Un sistema matemáticamente definido es una transformación que en forma única traza un patrón de entrada en un patrón de salida. Como se muestra en la *figura1*, cuando la entrada al sistema es denotada por el vector X y la salida denotada por el vector Y, la relación entrada-salida puede ser escrita como Y=*f*(X,W), donde W denota los pesos de la red. Los pesos y la estructura de los nodos interconectados en el sistema definen la transformación de entrada-salida desarrollado por la red. [23]

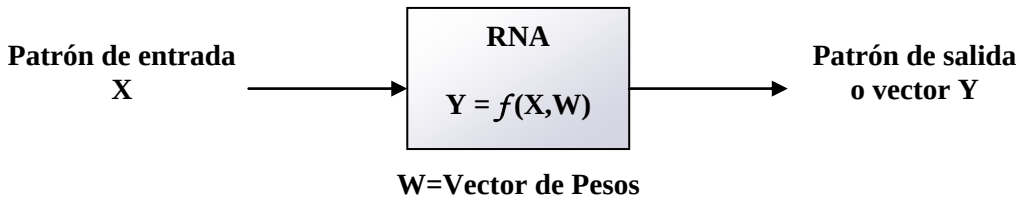


Figura III.1. ***La Red Neuronal Artificial representada como un sistema no lineal.***

Las RNA son modelos estadísticos no lineales. Una Red Neuronal es una regresión de dos - fases o un modelo de clasificación, y estas son generalmente representadas por diagramas. La red mas usada es la red Back-Propagation (Propagación hacia atrás o Retropropagación). Back-Propagation es un tipo de red de aprendizaje supervisado, el cual emplea un ciclo de propagación – adaptación de dos fases.

Las RNA constituyen una nueva técnica no paramétrica de análisis de datos multivariante, ya que no requiere de supuestos. Esta herramienta es más flexible y permite formular relaciones más complejas que las técnicas estadísticas tradicionales. [24]

3.2.2 Componentes de una Red Neuronal

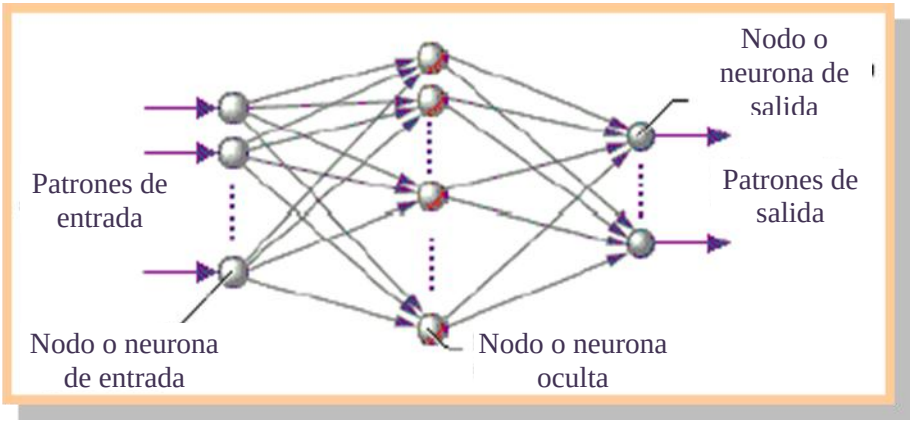


Figura III.2. ***Esquema de una RNA***

La RNA está constituida por neuronas interconectadas y arregladas por capas (de entrada, oculta y de salida). Los datos ingresan en la capa de entrada, pasan a través de la capa oculta y salen por la capa de salida. La capa oculta puede estar constituida por varias capas, sino hubiese capa oculta es como si se trabajara con el Perceptron Simple. Al modelo de la neurona se le llama habitualmente nodo o unidad.

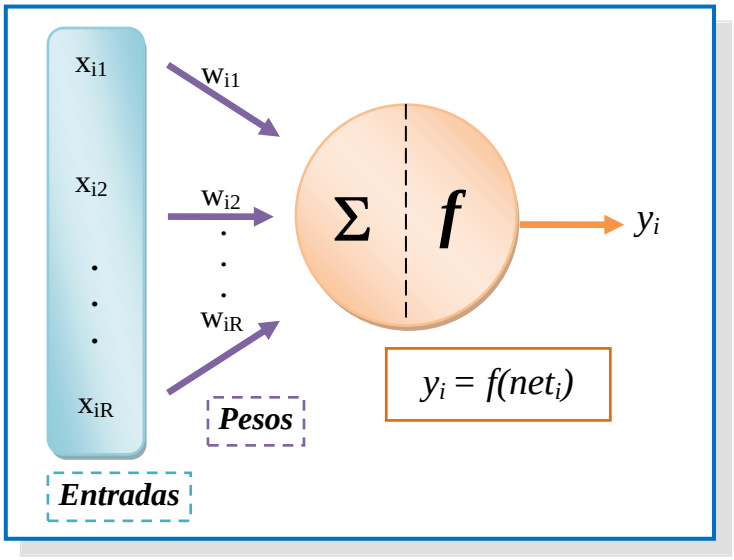


Figura III.3. ***Componentes de una RNA*.**

a. Entradas

Los patrones de entrada son las variables independientes denotadas como $\mathbf{X_j}$, las cuales ingresan a la neurona j, donde j = 1,2,…, R, y R es el número de variables independientes; cada registro es denotado como $\mathbf{X_i} = (X_{i1}, X_{i2},\ldots, X_{iR})$ es un vector de orden 1 x R, donde i = 1, 2,…., n, y n es el número de registros o individuos.

b. Pesos

Generalmente una neurona recibe muchas y múltiples entradas simultáneas. Cada entrada tiene su propio peso relativo (w_{ij}) el cual proporciona la importancia de la entrada dentro de la función de agregación de la neurona. Estos pesos realizan la misma función que realizan las fuerzas sinápticas de las neuronas biológicas. En ambos casos, algunas entradas son más importantes que otras de manera que tienen mayor efecto sobre el procesamiento de la neurona al combinarse para producir la respuesta neuronal.

Los pesos son coeficientes que pueden adaptarse dentro de la red que determinan la intensidad de la señal de entrada registrada

por la neurona artificial. Ellos son la medida de la fuerza de una conexión de entrada. Estas fuerzas pueden ser modificadas en respuesta de los ejemplos de entrenamiento de acuerdo a la topología específica o debido a las reglas de entrenamiento.

3.2.3 Estructura de una Red Neuronal

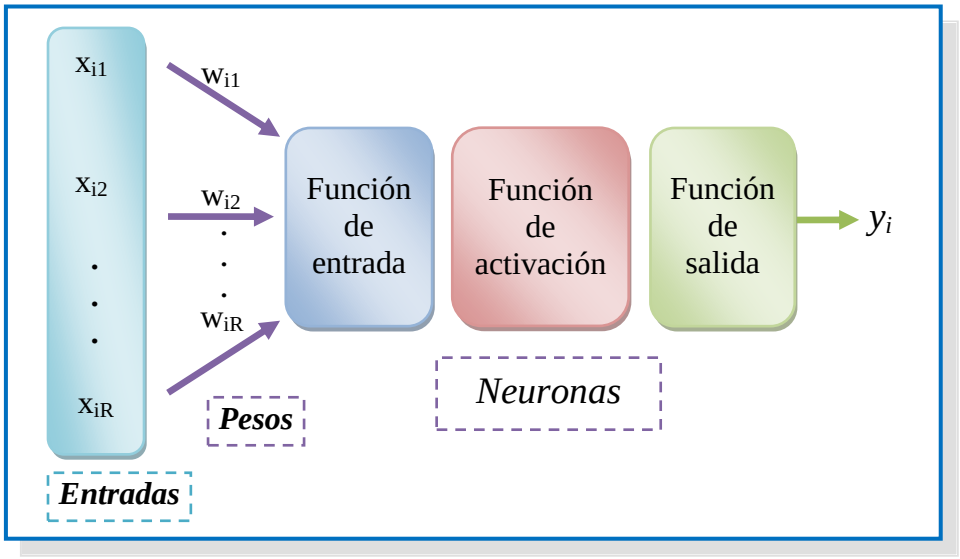


Figura III.4. ***Ejemplo de una neurona con R entradas y 1 salida.***

a. Función de entrada o de propagación

Esta regla permite obtener, a partir de las entradas y los pesos, el valor de la entrada neta o potencial post-sináptico "h_i" de la neurona: [25]

$$h_i(t) = \sigma_i(w_{ij}, x_j) \qquad \text{(III.5)}$$

La función más utilizada es la suma ponderada de todas las entradas, es decir se agrupan las entradas y pesos en dos vectores (x_1, x_2, ... , x_R) y (w_{1j}, w_{2j}, ... , w_{Rj}), con este se calcula la suma realizando el producto escalar sobre los dos vectores denominado también función lineal del tipo hiperplano.

$$h_i(t) = \sum_{j=1}^{R} w_{ij} \cdot x_j \qquad \text{(III.6)}$$

La función de propagación puede ser más compleja que simplemente una suma de productos. Las entradas y los pesos pueden ser combinados de diferentes maneras antes de pasarse el valor a la función de activación. Por ejemplo en esta función se puede usar: el mínimo, máximo, la mayoría, producto, o diversos algoritmos de normalización. El algoritmo específico para la propagación de las entradas neuronales está determinado por la elección de la arquitectura.

b. *Función de activación o transferencia*

Esta regla se encuentra en función de la suma ponderada de los registros ingresados y de los sesgos (*bias*), denotándose de la siguiente forma:

$$a_i(t) = f_i(b_i, h_i(t)) \qquad \text{(III.7)}$$

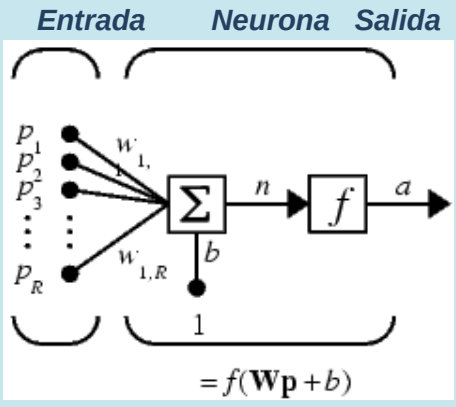


Figura III.8. ***Esquema de la función de transferencia.***

En nuestro caso se utilizará

$$a_i(t) = f_i(h_i(t) + b_i)$$

$$net_i(t) = h_i(t) + b_i$$

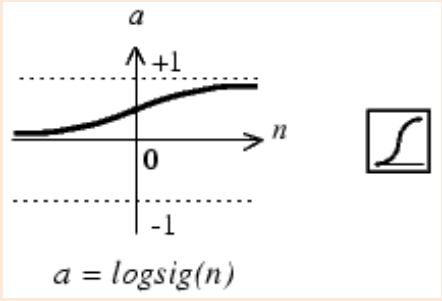
Donde:
p = X = vector de entrada (matriz de datos)
R = número de variables
W = Pesos
h = W x p
b = bias

a = función de transferencia

Algunas funciones de transferencia más utilizadas:

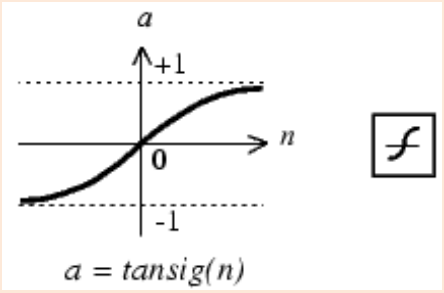
LOGSIG

La función ***logsig*** genera salidas entre 0 y 1 tal como las entradas van de infinito positivo y negativo.



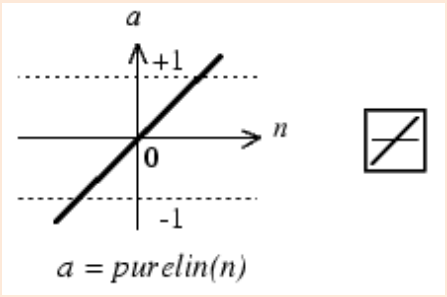
TANSIG

La función ***tansig*** genera salidas entre -1 y 1, y es tan bueno como el logsig.



PURELIN

La función lineal de transferencia ***purelin*** genera salidas que van de $\langle -\infty, \infty \rangle$.



ESCALÓN

La función escalón se asocia a redes cuya salida es binaria, en las cuales cuando la Net es mayor o igual que cero, la activación es 1. Si la Net es menor que 0, la activación es 0.

$$g(Net) = \begin{cases} 1 & si \quad Net \geq 0 \\ 0 & si \quad Net < 0 \end{cases}$$

Figura III.9. ***Gráficas de algunas funciones de transferencia*.**

En la función de activación el valor de la salida de combinación puede ser comparada con algún valor umbral para determinar la salida de la neurona. Si la suma es mayor que el valor umbral, neurona generará una señal. Si la suma es menor que el valor umbral, ninguna señal será generada. Normalmente el valor umbral, o valor de la función de transferencia, es normalmente no lineal.

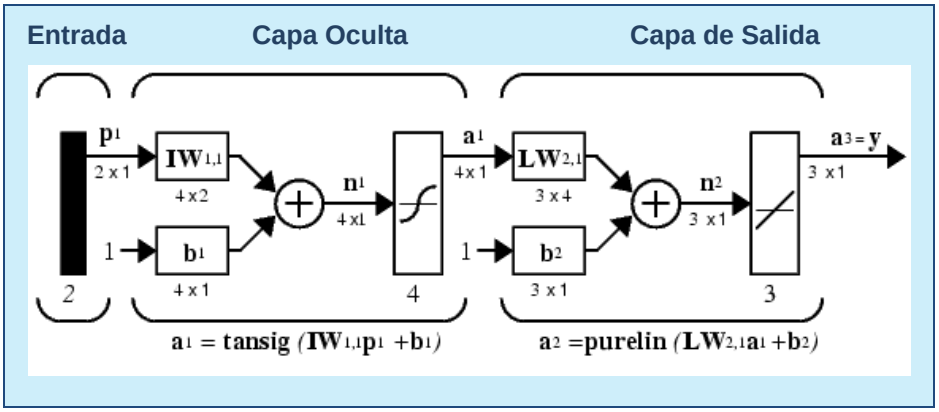


Figura III.10. ***Ejemplo de función de transferencia en la capa oculta y en la capa de salida.***

c. Función de salida (Competitividad)

Cada elemento de procesamiento tiene permitido una única salida $y_i(t)$ que puede estar asociada con un número elevado de otras neuronas. Usualmente, la salida es directamente equivalente al valor resultante de la función de activación.

$$y_i(t) = F_i(a_i(t)) = a_i(t) \qquad \text{(III.11)}$$

Algunas topologías de redes neuronales, sin embargo, modifican el valor de la función de transferencia para incorporar un factor de competitividad entre neuronas que sean vecinas. Las neuronas tienen permitidas competir entre ellas, inhibiendo a otras neuronas a menos que tengan una gran fortaleza.

d. Función de error y el valor propagado hacia atrás

En la mayoría de algoritmos de entrenamiento de redes neuronales necesitamos calcular la diferencia entre la salida actual y la esperada. Esta diferencia es transformada por la función de error correspondiente a la arquitectura particular.

El error de la neurona se propaga normalmente dentro del algoritmo de aprendizaje de otra neurona. Este término de error es algunas veces denominado error actual. El error actual es propagado hacia atrás a la capa anterior, siendo este valor o bien el valor actual de error de esa capa obtenido al escalarlo de alguna manera (lo habitual es usando la derivada de la función de transferencia) o bien es tomado como la salida esperada (esto sucede en algunas topologías).

Normalmente el valor que se propaga hacia atrás, una vez escalado por la función de aprendizaje, se multiplica por los pesos de la neurona para modificarlas antes de pasar al ciclo siguiente. [26]

3.2.4 Escalamiento y Limitación

El valor de salida de la función de activación puede ser procesado de manera adicional mediante un escalamiento y limitación. El escalamiento simplemente multiplica el valor de la función de transferencia por un factor de escala y después se le suma un desplazamiento.

El mecanismo de limitación es el que asegura que el resultado del escalamiento no excede una cota superior o inferior. Esta limitación se realiza de manera adicional a los límites que puede imponer la función de transferencia original.

Normalmente este tipo de escalamiento y limitación es usado principalmente en topologías usadas para verificar modelos neuronales biológicos. [26]

3.2.5 Funcionamiento de una Red Neuronal

En la ***figura III.12*** se describe el procedimiento para usar redes neuronales. Originalmente la red neuronal no dispone de ningún tipo de conocimiento útil almacenado. Para que la red neuronal ejecute una tarea es preciso entrenarla, lo que en estadística diríamos ***estimar los parámetros***.

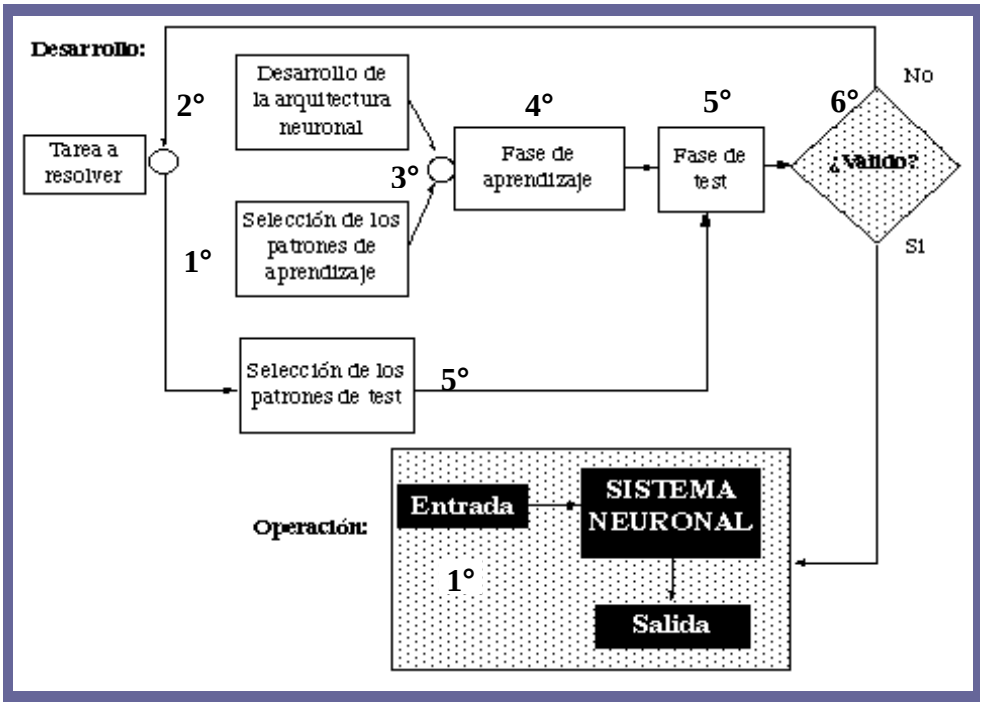


Figura III.12. ***Modo de trabajo con redes neuronales***

Procedimiento Estadístico

El procedimiento estadístico en la *figura III.12* sería:

1º Seleccionar un conjunto de datos, o patrones de aprendizaje en términos de RNA.
2º Desarrollar la arquitectura neuronal, número de neuronas, tipo de red.
3º Seleccionar el modelo y los números de variables dependientes e independientes.
4º Se procede a la fase de aprendizaje o estimación del modelo.
5º Se procede a la fase de prueba, con la cual se evaluará el modelo.
6º Se validan los resultados.

3.3 Clasificación :

Los distintos modelos de redes neuronales pueden clasificarse de acuerdo su naturaleza, topología, mecanismo de aprendizaje y tipo de asociación.

3.3.1 La naturaleza de las señales de entrada y salida, con esto las redes neuronales pueden clasificarse en: [25]

i. *Analógicas:* Las redes analógicas suelen presentar funciones de activación continuas, habitualmente lineales o sigmoides. Entre estas redes neuronales destacan las redes de Backpropagation, la red continua de Hopfield, la de Contrapropagación, la Memoria Lineal Asociativa, la Brain-State-in-Box, y los modelos de Kohonen (mapas auto-organizados (S.O.M.) y Learning Vector Quantizer, (L.V.Q.).

ii. *Discretas (Binarias):* procesan datos de naturaleza discreta, para acabar emitiendo una respuesta discreta. Entre las redes binarias destacan la Maquina de Boltzman, la Máquina de Cauchy, la red discreta de Hopfield, el Cognitrón y el Neogognitrón.

iii. *Híbridas:* procesan entradas analógicas para dar respuestas binarias. Entre ellas destacan el Perceptrón, la red Adaline y la Madaline.

3.3.2 La topología de la red, donde las redes pueden clasificarse de acuerdo al número de capas (niveles de neuronas). [25]

Se denomina arquitectura a la topología, estructura o patrón de conexionado de una red neuronal. En una RNA los nodos se conectan por medio de sinapsis, esta estructura de conexiones sinápticas determina el comportamiento de la red.

En general, las neuronas se suelen agrupar en unidades estructurales que se denominan capas. Finalmente, el conjunto de una o más capas constituye la red neuronal. Se distinguen tres tipos de capas:

- *De entrada*: Una capa de entrada o sensorial está compuesta por neuronas que reciben datos o señales procedentes del entorno.
- *Oculta*: es aquella que no tiene conexión directa con el contorno.
- *De salida*: es aquella cuyas neuronas proporcionan la respuesta de la red neuronal.

a. ***Tipos de arquitecturas neuronales:***

- *Redes monocapa:* son aquellas compuestas por una capa de neuronas, que intercambian señales con el exterior y que constituyen a un tiempo la entrada y salida del sistema.

- *Redes multicapa (layered networks):* son aquellas cuyas neuronas se organizan en varias capas, disponen de conjuntos de neuronas jerarquizadas en distintos niveles, con al menos una capa de entrada y otra de salida, y una o varias capas intermedias.

3.3.3 El mecanismo de aprendizaje, es el proceso donde una red neuronal adquiere la capacidad de desempeñar funciones específicas dependiendo del problema que se pretende abordar. [27]

En el proceso de entrenamiento se modifican los pesos que afectan a las entradas de la neurona. Este proceso de cambio en los pesos de las conexiones de entradas para conseguir un valor deseado también es llamado función de adaptación. Las RNA pueden ser entrenadas mediante:

a. *Entrenamiento Supervisado*.- mediante este tipo se introduce a la red una serie de patrones de entrada y salida. La red es capaz de ajustar los pesos con el fin de memorizar la salida deseada.

   La mayor parte de arquitecturas de RNA son entrenadas mediante métodos supervisados. En este tipo de entrenamiento, la salida de la RNA es comparada con al valor deseado de salida. Los pesos, que normalmente han sido establecidos de manera aleatoria en un principio, son ajustados por la red de manera que en la siguiente iteración, también denominado ciclo, producirá un resultado más cercano entre el valor esperado y la salida real.

   Una vez que el entrenamiento termina los pesos se fijan, aunque algunas redes permiten el entrenamiento continuo pero con una tasa de aprendizaje baja. Esto ayuda a la red a adaptarse de manera gradual a situaciones de cambio. El conjunto de patrones

de entrenamiento necesita ser lo suficientemente grande como para contener toda la información necesaria para que la red aprenda todas las relaciones que son importantes. No sólo los conjuntos de patrones de entrenamiento tienen que ser grandes sino que las sesiones de entrenamiento deben incluir una gran cantidad y variedad de datos.

Si la red es entrenada con un ejemplo cada vez, todos los pesos son establecidos de manera demasiado meticulosa, para cada hecho puede ser drásticamente alterado por el entrenamiento para el siguiente hecho, esto puede provocar que la red se olvide de algunos hechos durante el entrenamiento para aprender otros. Como resultado, el sistema tiene que aprender todo a la vez, buscando la mejor combinación de pesos para todos los hechos.

b. *Entrenamiento No supervisado*.- aquí la red responde clasificando los patrones de entrada en función de las características más adecuadas de cada uno.
   Actualmente este tipo de entrenamiento está limitado a las redes conocidas como mapas autoorganizados (Kohonen maps, SOM). Aún están en proceso de estudio ya que su funcionamiento no es del todo conocido.

c. *Entrenamiento Autosupervisado*.- en este tipo la propia red corrige los errores en la interpretación a través de una realimentación.

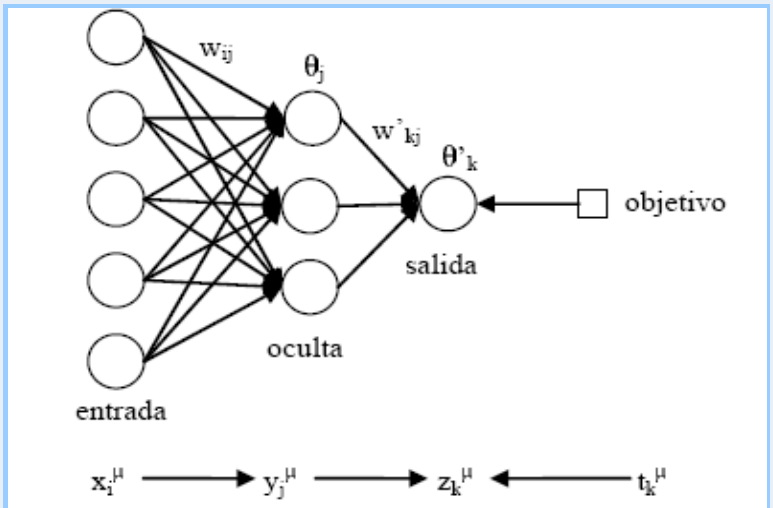
d. *Redes híbridas:* son un enfoque mixto en el que se utiliza una función de mejora para facilitar la convergencia. Un ejemplo de este último tipo es la red de base radial.

3.3.4 El tipo de asociación de las señales de entrada y salida y la forma de representar estas señales. [25]

Las conexiones entre las neuronas pueden ser excitatorias o inhibitorias, es decir cuando el peso sináptico es negativo define una conexión inhibitoria, mientras que uno positivo determina una conexión excitatoria.

Las conexiones intra-capa, también denominadas laterales, tienen lugar entre las neuronas pertenecientes a una misma capa, mientras que las conexiones inter-capa se producen entre las neuronas de las diferentes capas. Existen además conexiones realimentadas, que tienen un sentido contrario al de entrada-salida. En algunos casos puede existir realimentación incluso de una neurona consigo misma.

Generalmente (aunque no en todos los modelos), una vez que el sistema ha sido entrenado, el aprendizaje “se desconecta”, por lo que los pesos y la estructura quedan fijos, estando la red neuronal ya dispuesta para procesar datos. Este modo de operación también es denominado “recall”.



Donde:

X = Matriz de datos (p).

Y = 1° Función de Transferencia.

Z = 2° Función de Transferencia o función de salida.

***T = Vector de los datos esperados*.**

Figura III.13. ***Gráfica del tipo de asociación de una RNA***

Entre dos capas de neuronas existe una red de pesos de conexión, que puede ser de los siguientes tipos: hacia delante, hacia atrás, lateral y de retardo.

a. *Conexiones hacia delante (feedforward):* los valores de las neuronas de una capa inferior son propagados hacia las neuronas de la capa superior por medio de las redes de conexiones hacia delante, la información circula en un único sentido desde las neuronas de entrada a las de salida.

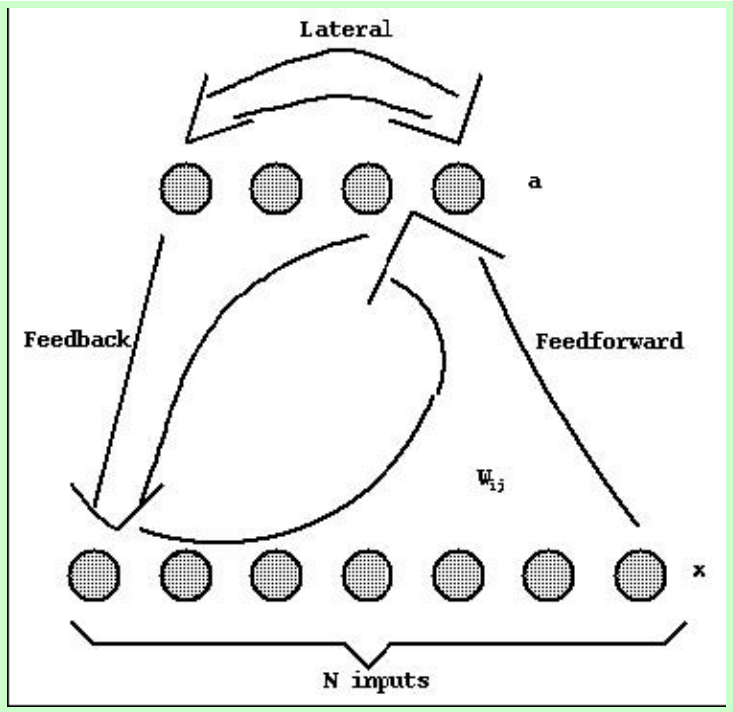


Figura III.14: ***Esquema de conexiones*****.**

b. *Conexiones hacia atrás (feedback):* estas conexiones llevan los valores de las neuronas de una capa superior a otras de la capa inferior, la información puede circular entre las capas en cualquier sentido.

c. *Conexiones laterales:* Un ejemplo típico de este tipo es el circuito “el ganador toma todo” (winner-takes-all), que cumple un papel importante en la elección del ganador: a la neurona de salida que da el valor más alto se le asigna el valor total (por ejemplo, 1), mientras que a todas las demás se le da un valor de 0.

d. *Conexiones con retardo:* los elementos de retardo se incorporan en las conexiones para implementar modelos dinámicos y temporales, es decir, modelos que precisan de memoria.

3.4 Modelos

Las redes neuronales tienen una serie de modelos tales como [28]:

- Perceptron
- Adaline
- Perceptron multicapa
- Memorias asociativas
- Máquina de Bolzman
- Máquina de Cauchy
- Propagación hacia atrás (Backpropagation)
- Redes de Elman
- Redes de Hopfield
- Red de contrapropagación
- Redes de neuronas de base radial
- Redes de neuronas de aprendizaje competitivo
- Mapas Autoorganizados (RNA)
- Crecimiento dinámico de células
- Gas Neuronal Creciente
- Redes ART (*Adaptative Resonance Theory*)

3.4.1 Fases en la modelización con las redes neuronales:

Fase de entrenamiento: se usa un conjunto de datos o patrones de entrenamiento para determinar los pesos y bias (parámetros) que definen el modelo de red neuronal; la red puede ser entrenada por funciones de aproximación (regresión no lineal), modelos de asociación y modelos de clasificación. El proceso de entrenamiento requiere de un conjunto apropiado de ejemplos para el comportamiento de la red ingresando para ello la matriz de datos (p) y las salidas esperadas (t). Durante el entrenamiento los pesos y bias de la red son ajustadas iterativamente para minimizar la función de error. Por defecto la función de error que la red utiliza es el ***cuadrado***

medio del error (MSE), es decir el promedio de los errores al cuadrado.

Se tiene varios algoritmos para el entrenamiento de la red, la mayoría de estos algoritmos usan el gradiente de la función del error para determinar el ajuste de los pesos y minimizar el error cometido entre la salida obtenida por la red neuronal y la salida deseada. El gradiente es determinado usando la técnica llamada backpropagation, la cual involucra el computo hacia atrás a través de la red. El cómputo de backpropagation es derivada de la utilización de la regla de la cadena de Cálculo. El entrenamiento básico del algoritmo backpropagation consiste en que los pesos se mueven en dirección de la pendiente negativa.

Fase de Prueba: en la fase anterior, el modelo puede que se ajuste demasiado a las particularidades presentes en los patrones de entrenamiento, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos (sobreajuste). Para evitar el problema del sobreajuste, es aconsejable utilizar un segundo grupo de datos diferentes a los de entrenamiento, el grupo de validación, que permita controlar el proceso de aprendizaje.

Generalmente, los pesos óptimos se obtienen optimizando (minimizando) alguna función. Por ejemplo, un criterio muy utilizado en el llamado entrenamiento supervisado, es minimizar el error cuadrático medio entre el valor de salida y el valor real esperado.

3.5 Perceptron Multicapa

Un Perceptron multicapa o multinivel es una red de tipo *feedforward* compuesta de varias capas de neuronas entre la entrada y la salida de la misma. Es llamada también red de alimentación hacia delante o red de propagación hacia atrás. Esta red permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como lo hace el Perceptron de un solo nivel. [18]

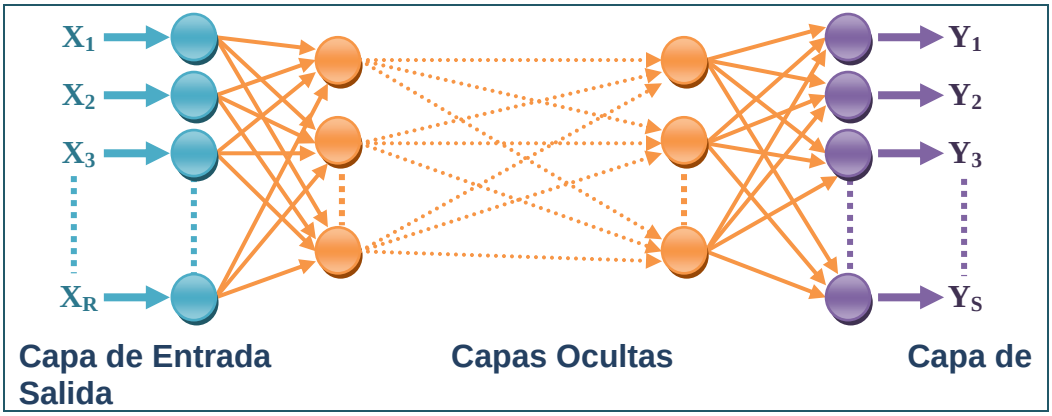


Figura III.15: ***Perceptron Multicapa (red feedforward multicapa).***

El Perceptron básico de dos capas (la de entrada con neuronas lineales y la de salida con función de activación de tipo escalón) sólo puede establecer dos regiones separadas por una frontera lineal en el espacio de patrones de entrada. Un Perceptron de tres niveles de neuronas puede formar cualquier región convexa en este espacio. Las regiones convexas se forman mediante la intersección entre las regiones formas por cada neurona de la segunda capa. Cada uno de estos elementos se comporta como un Perceptron simple, activándose su salida para los patrones de un lado del hiperplano. Si el valor de los pesos de las conexiones entre las L neuronas de la segunda capa y una neurona del nivel de salida (S=1) son todos uno y el umbral de la de salida es L – a, donde 0 < a < 1, entonces la salida de la red se activará sólo si las salidas de todos los nodos de la segunda capa están activos. Esto equivale a ejecutar la operación lógica AND en el nodo de salida, resultando una región de decisión intersección de todos los semiplanos formados en el nivel anterior. La región de decisión resultante de la intersección serán regiones convexas con un número de lados a lo sumo igual al número de neuronas de la segunda capa. [18]

Este análisis presenta el problema de selección del número de neuronas ocultas de un Perceptron de tres capas. En general, este número deberá ser lo suficientemente grande como para que se forme una región lo suficientemente compleja para la resolución del problema. Sin embargo, tampoco es conveniente que el número de neuronas sea tan grande que la estimación de los pesos sea fiable para el conjunto de patrones de entrada disponible.

Rumelhart, Hinton y Williams (1986) formalizaron un método para que una red del tipo perceptrón multicapa aprendiera la asociación que existe entre un conjunto de patrones de entrada y sus salidas correspondientes. Este método, conocido como backpropagation error (propagación del error hacia atrás), también denominado método de gradiente decreciente, ya había sido descrito anteriormente por Werbos (1974), Parker (1982) y Le Cun (1985), aunque fue el Parallel Distributed Processing Group (grupo PDP) Rumelhart y colaboradores, quien realmente lo popularizó. La importancia de la red backpropagation consiste en su capacidad de organizar una representación interna del conocimiento en las capas ocultas de neuronas, a fin de aprender la relación que existe entre un conjunto de entradas y salidas. Posteriormente, aplica esa misma relación a nuevos vectores de entrada con ruido o incompletos, dando una salida activa si la nueva entrada es parecida a las presentadas durante el aprendizaje.

Esta característica importante es la capacidad de generalización, entendida como la facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento.

3.5.1 Algoritmo Backpropagation

Es un algoritmo de la gradiente descendente, como la regla de aprendizaje Widrow-Hoff, en el cual los pesos de la red se mueven en dirección de la gradiente negativa de la función del error. El término backpropagation se refiere a la manera en que la gradiente es calculada por las redes multicapa no lineales. Existen variaciones en el algoritmo básico que es basado en otras técnicas de optimización normales, como la gradiente conjugada y los métodos de Newton. Este algoritmo tiene por objetivo minimizar el error total cometido por la red, definido como la suma de los cuadrados de los errores cometidos. [29]

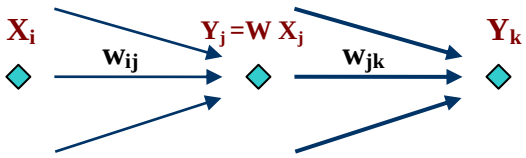
Pasos del Algoritmo Backpropagation

1. Calcular el error en la salida para cada patrón.
2. Ajustar los pesos en la capa de salida para reducir el error.
3. Propagar los errores hacia la capa de entrada, ajustando los pesos de las capas ocultas.
4. Se repiten los anteriores pasos en forma iterativa.
5. Los pesos pueden ser actualizados de dos formas:
   - Al presentar cada patrón.
   - Al presentar el conjunto de entrenamiento total.

Características

- El algoritmo busca el mínimo de la función error a partir de un conjunto de patrones de entrenamiento.
- El algoritmo precisa que la función de activación sea diferenciable.
- Entrenar consiste en modificar los pesos de la red. Estos se modifican hacia la dirección descendente de la función del error.
- La red entrenada es capaz de generalizar, clasificando correctamente patrones ruidosos o incompletos.

Diagrama de Partida



Las X's son las variables de entrada, las Y's son las variables de salida, y las W son los pesos de las conexiones.

Definición de las entradas y salidas

Sean: net_j = Entrada neta a la unidad j

net_k = Entrada neta a la unidad j

w_{ij} = Peso de la unidad i a la j

La entrada neta a las distintas unidades es:

$$net_j = \sum_i x_i \cdot w_{ij} \qquad net_k = \sum_j x_j \cdot w_{jk}$$

La salida de las distintas unidades es:

$$o_j = x_j = F(net_j) \qquad o_k = F(net_k)$$

Definición del error

Sean: E = Error total cometido

t_k = Salida deseada

o_k = Salida obtenida

El error total cometido será:

$$E = \frac{1}{2}\sum_k (t_k - o_k)^2 = \frac{1}{2}\sum_k (\quad)^2 \qquad \text{(III.16)}$$

El algoritmo Backpropagation se basa en el método del gradiente, donde los pesos se modifican mediante el siguiente proceso iterativo.

$$w_{ij} = w_{ij} - \mu \cdot \frac{\partial E}{\partial w_{ij}} \qquad \text{(III.17)}$$

Definición de δ

- Utilizaremos un término δ $\qquad \delta_j = \dfrac{\partial E}{\partial net_j}$ $\qquad$ (III.18)
- Para las unidades de salida $\dfrac{\partial E}{\partial w_{jk}} = \dfrac{\partial E}{\partial net_k} \cdot \dfrac{\partial net_k}{\partial w_{jk}} = \delta_k \cdot x_j$ (III.19)

- Para las unidades ocultas $$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ij}} = \delta_j \cdot x_i \qquad \text{(III.20)}$$

- Deduciremos el valor de δ de forma separada para las unidades de salida y ocultas.

Unidades de Salida

- Utilizando la Regla de la cadena

$$\delta_k = \frac{\partial E}{\partial net_j} = \frac{\partial E}{\partial e_k} \cdot \frac{\partial e_k}{\partial o_k} \cdot \frac{\partial o_k}{\partial net_j} = e_k \cdot (-1) \cdot F'(net_k) \qquad \text{(III.21)}$$

- Por tanto, la actualización de los pesos es:

$$w_{jk} = w_{jk} - \mu \cdot \frac{\partial E}{\partial w_{jk}} = w_{jk} - \mu \cdot \delta_k \cdot x_j$$

$$w_{jk} = w_{jk} + \mu \cdot (t_k - o_k) \cdot F'(net_k) \cdot x_j \qquad \text{(III.22)}$$

Unidades Ocultas

- Utilizando la Regla de la cadena

$$\delta_j = \sum_k \left(\frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial o_j} \right) \cdot \frac{\partial o_j}{\partial net_j} = \sum_k (\delta_k \cdot w_{jk}) \cdot F'(net_j) \qquad \text{(III.23)}$$

- Por tanto, la actualización de los pesos es:

$$w_{ij} = w_{ij} - \mu \cdot \frac{\partial E}{\partial w_{ij}} = w_{ij} - \mu \cdot \delta_j \cdot x_i$$

$$w_{ij} = w_{ij} - \mu \cdot \delta_j \cdot x_i \qquad \text{(III.24)}$$

Algoritmos de aprendizaje en Backpropagation

Hay dos maneras diferentes en el que este algoritmo de la gradiente descendente puede llevarse a cabo: del modo incremental y del modo de lote. En el modo incremental, la gradiente es calculada y los pesos se

actualizan después de que cada entrada se aplica a la red. En el modo de lote todas las entradas se aplican a la red antes de que los pesos se actualicen. [30]

a. ***Traingd:*** Algoritmo de pasos descendientes, que actualiza pesos y ganancias variándolos en la dirección negativa del gradiente de la función del error. Es un algoritmo de aprendizaje muy lento. Con este algoritmo el aprendizaje de la red se detendrá: si el número de iteraciones se excede, si se alcanzó el valor del error propuesto como meta, si la magnitud del gradiente es menor, o si el tiempo de entrenamiento excesivo.

b. ***Traingdm:*** Equivale al algoritmo tradicional, más un nuevo coeficiente de momentum, que interviene en el proceso de actualización de los pesos. Si el error de la red en una iteración dada, excede el valor del error en la iteración anterior, en un valor mayor al definido por un radio de cobertura dado el que puede determinarse por medio de una función y que está típicamente alrededor de 1.04, los nuevos pesos y ganancias son descartados y el coeficiente de momentum es fijado en cero.

c. ***Traingda:*** Algoritmo de Gradiente Descendiente, que emplea una tasa de aprendizaje adaptiva durante el proceso de entrenamiento. La tasa de aprendizaje varía entre 0.01 y 1, una tasa de aprendizaje muy pequeña torna lento el aprendizaje, pero si se incrementa demasiado el aprendizaje puede tornarse inestable y crear divergencia, por esto la función ***traingda*** varía la tasa de aprendizaje tratando de sacar provecho de la inclinación del gradiente en cada momento; su gran desventaja es que los pesos iniciales varían muy poco así se encuentren distantes de los valores de convergencia.

d. ***Trainrp:*** Las redes multicapa, utilizan típicamente una función de transferencia sigmoidal en las capas ocultas, estas funciones comprimen un infinito rango de entradas, dentro de un finito rango de salidas, además se caracterizan porque su pendiente tendera cada vez más a cero, mientras más grande sea la entrada que se le

presenta a la red, esto ocasiona problemas cuando se usa un algoritmo de entrenamiento de pasos descendientes, porque el gradiente empieza a tomar valores muy pequeños y por lo tanto no habrán cambios representativos en los pesos y las ganancias, así se encuentren bastante lejos de sus valores óptimos. El propósito del algoritmo Backpropagation Resileint (RPROP) es eliminar este efecto en la magnitud de las derivadas parciales. En este algoritmo solamente el signo de la derivada es utilizado para determinar la dirección de actualización de los parámetros, la magnitud de las derivadas no tiene efecto en la actualización. La magnitud en el cambio de cada peso es determinada por separado; el valor del incremento de pesos y ganancias es determinado por un factor, así la derivada parcial del error con respecto a los pesos tenga el mismo signo durante dos iteraciones sucesivas; el valor de decremento está determinado por otro factor así la derivada del error con respecto a los pesos haya cambiado de signo con respecto a la anterior iteración; si la derivada es cero, entonces el valor actualizado se conserva; si los pesos continúan cambiando en la misma dirección durante varias iteraciones, la magnitud de cambios de los pesos decrementan.

e. ***Trainbfg:*** Algoritmo alternativo que emplea la técnica del gradiente conjugado, su expresión matemática se deriva del método de Newton, con la ventaja de que no es necesario computar las segundas derivadas; este algoritmo requiere más capacidad de almacenamiento que el algoritmo tradicional, pero generalmente converge en menos iteraciones. Requiere de un cálculo aproximado de la matriz Hessiana, la cual es de dimensiones n^2 x n^2, donde n es la cantidad de pesos y ganancias de la red; para redes que involucren una gran cantidad de parámetros es preferible emplear el algoritmo trainrp.

f. ***Trainlm:*** Algoritmo que actualiza los pesos y las ganancias de acuerdo a la optimización de Levenberg-Marquardt. Es el algoritmo más rápido para redes Backpropagation; tiene la desventaja de requerir de un set de entrenamiento lo más estándar posible, pues

de otra forma solo aproximará correctamente valores que se encuentren dentro de los patrones de aprendizaje. Si el set de entrenamiento es muy extenso, se recomienda reducir el Jacobiano.

Procedimiento para entrenar la Red

El procedimiento básico para entrenar la red está plasmado en la siguiente descripción:

1. Se aplica un vector de entrada a la red, y se calculan los correspondientes valores de salida.
2. Se comparan las salidas obtenidas con las salidas correctas, y se determina una medida del error.
3. Se determina en que dirección (+ ó -) debe cambiar cada peso con objeto de reducir el error.
4. Se determina la cantidad en que es preciso cambiar cada peso
5. Se aplican las conexiones a los pesos.
6. Se repiten los pasos del 1 al 5 con todos los vectores de entrenamiento hasta que el error para todos los vectores del conjunto de entrenamiento quede reducido a un valor aceptable.

3.6 Aplicación de las Redes Neuronales

Redes Neuronales puede ser utilizado en dominios muy concretos, es decir problemas reales. Estas son aplicadas en muchas áreas como: Biología, Ingeniería, Economía, Estadística, etc.

Los campos de aplicación de las redes neuronales son habitualmente todos aquellos en los que se utilizan o pueden utilizarse modelos estadísticos y/o lineales. En general la utilización de las redes neuronales proporciona resultados mucho mejores. Las RNA pueden utilizarse en un gran número y variedad de aplicaciones, tanto comerciales como militares. Hay muchos tipos diferentes de redes neuronales; cada uno de los cuales tiene una aplicación particular más apropiada. Algunos campos donde se aplican las redes neuronales: [31]

- **Biología:**
    - Aprender más acerca del cerebro y otros sistemas.
    - Obtención de modelos de la retina.

- **Empresa:**
    - Evaluación de probabilidad de formaciones geológicas y petrolíferas.
    - Identificación de candidatos para posiciones específicas.
    - Explotación de bases de datos.
    - Optimización de plazas y horarios en líneas de vuelo.
    - Reconocimiento de caracteres escritos.
    - Modelado de sistemas para automatización y control.

- **Negocios**
    - Marketing
    - Venta cruzada
    - Campanas de venta

- **Medio ambiente:**
    - Analizar tendencias y patrones.
    - Previsión del tiempo.

- **Finanzas:**
    - Previsión de la evolución de los precios.
    - Valoración del riesgo de los créditos.
    - Identificación de falsificaciones o fraudes.
    - Interpretación de firmas.
    - Predicción de índices.
    - Predicción de la rentabilidad de acciones.

- **Manufacturación:**
    - Robots automatizados y sistemas de control (visión artificial y censores de presión, temperatura, gas, etc.).
    - Control de producción en líneas de procesos.
    - Inspección de la calidad.

- **Medicina:**
    - Analizadores del habla para ayudar en la audición de sordos profundos.
    - Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos(electrocardiograma, encefalogramas, análisis sanguíneo).
    - Monitorización en cirugías.
    - Predicción de reacciones adversas en los medicamentos.
    - Entendimiento de la causa de los ataques cardíacos.
    - Análisis de Imágenes.
    - Distribución de recursos.

- **Militares:**
    - Clasificación de las señales de radar.
    - Creación de armas inteligentes.
    - Optimización del uso de recursos escasos.
    - Reconocimiento y seguimiento en el tiro al blanco.

- **Tratamiento de textos y proceso de formas.**
    - Reconocimiento de caracteres impresos mecánicamente.
    - Reconocimiento de gráficos.
    - Reconocimiento de caracteres escritos a mano.
    - Reconocimiento de escritura manual cursiva.

- **Alimentación**
    - Análisis de olor y aroma.
    - Perfilamiento de clientes en función de la compra.
    - Desarrollo de productos.
    - Control de Calidad.

- **Energía.**
    - Predicción consumo eléctrico
    - Distribución recursos hidráulicos para la producción eléctrica
    - Predicción consumo de gas ciudad

- **Ciencia e Ingeniería.**
    - Análisis de datos y clasificación
    - Ingeniería Química.
    - Ingeniería Eléctrica.

- **Transportes y Comunicaciones.**
    - Optimización de rutas.
    - Optimización en la distribución de recursos

Desde el punto de vista de los casos de aplicación, la ventaja de las redes neuronales reside en el procesado paralelo, adaptativo y no lineal. El dominio de aplicación de las redes neuronales también se lo puede clasificar de la siguiente forma: asociación y clasificación, regeneración de patrones, regresión y generalización, y optimización.

3.7 Ventajas

- La capacidad del aprendizaje adaptativo, como las redes neuronales pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos a priori ni de especificar funciones de distribución de probabilidad. [18]

- La autoorganización, la cual consiste en la modificación de la RNA completa para llevar a cabo un objetivo específico. Ocasionando la generalización de la facultad de la RNA para responder apropiadamente cuando se le presentan datos o situaciones a las que no había sido expuesta antes. [18]

- Tolerancia a fallos respecto a los datos es cuando las RNA pueden aprender a reconocer patrones con ruido, distorsionados o incompletos; tolerancia a fallos respecto a la estructura de la red, las RNA pueden seguir realizando su función con cierta degradación aunque se destruya la red. [18]

3.8 Desventajas

- Determinar el número adecuado de neuronas ocultas, de incurrir a problemas de memorización o a que la red no desempeñe sus tareas. [32]

- La introducción de variables irrelevantes o que varíen entre sí, ocasionando un sobreajuste innecesario en el modelo, esto origina la disminución de la capacidad de generalización del modelo. [33]

CAPÍTULO IV:

Aplicación

4.1 Introducción

En este capítulo se desarrolló la comparación de los métodos de clasificación: Análisis de Regresión Logística, Árboles de Clasificación y Redes Neuronales, utilizando una base de datos de un Banco Alemán. La comparación se realizó en función a los modelos obtenidos mediante estos métodos, determinando el mejor método en base a su poder de clasificación y predicción para el caso de Riesgo Crediticio. Para la ejecución de estos métodos se empleo el programa TANAGRA v.1.4.2.

4.2 Descripción de los Datos:

Los datos a usarse en éste trabajo provienen del *"Institut f"ur Statistik und "Okonometrie Universit"at Hamburg FB Wirtschafts wissenschaften Von-Melle-Park 5 2000 Hamburg 13 - Professor Dr. Hans Hofmann"*, corresponden a un grupo de personas que solicitaron un préstamo en un Banco Alemán.

El objetivo de este trabajo es evaluar la clasificación de un cliente en base al préstamo que se le otorgó en el banco y determinar si un nuevo cliente que solicité un préstamo es un buen o mal pagador.

La base de datos contiene 1000 registros y 20 variables, de las cuales 13 son categóricas y 7 cuantitativas.

4.3 Descripción de las variables:

Tabla IV.1: Descripción de las variables

Nº	VARIABLES	CATEGORÍAS
1	Estado de cuenta corriente existente	0. Ninguna Cuenta Corriente 1. < 0 DM. 2. [0, 200 > DM 3. >= 200 DM
2	Duración del crédito en meses	------
3	Historial Crediticio	0. Cuenta crítica / otros créditos en otro banco. 1. Retraso de pago en el pasado. 2. Créditos existentes devueltos debidamente hasta ahora. 3. Todos los créditos en este banco fueron devueltos. 4. Ningún crédito / todos devueltos debidamente.
4	Objetivo del crédito	1. Auto nuevo 2. Auto usado 3. Muebles/equipo 4. Radio/televisión 5. Utensilios domésticos 6. Reparación 7. Educación 8. Vacaciones 9. Reciclaje 10. Negocio 11. Otros
5	Cantidad de crédito	------
6	Cuenta de ahorros	0. Ninguna Cuenta de ahorros. 1. < 100 DM 2. [100 , 500 > DM 3. [500 , 1000 > DM 4. >= 1000 DM
7	Tiempo de Empleado	0. Desempleado 1. < 1 año 2. [1, 4 > años 3. [4, 7 > años 4. >= 7 años
8	Tasa de amortización	1, 2, 3, 4
9	Estado civil y sexo	1. Hombre: divorciado / separado 2. Mujer: divorciada / separada / casada 3. Hombre: Soltero 4. Hombre: casado / viudo 5. Mujer: soltera
10	Otros deudores ó garantes	0. Ninguno 1. Co-aspirante 2. Garante
11	Tiempo de Residencia	1, 2, 3, 4
12	Propiedad	0. Ninguna propiedad 1. Coche u otro 2. Acuerdo de ahorros/ seguro de vida

		3. Bienes inmuebles
Nº	**VARIABLES**	**CATEGORÍAS**
13	Edad	------
14	Otros planes de financiación	0. Ninguno 1. Tiendas 2. Banco
15	Alojamiento	0. Gratis 1. Propio 2. Alquiler
16	Número de créditos en este banco	1, 2
17	Trabajo	0. Desempleado / inexperto 1. Inexperto 2. Empleado experto 3. Empleado altamente calificado
18	Número de personas que están obligadas a proporcionar mantenimiento	1, 2
19	Teléfono	0. No 1. Si
20	Trabajador Extranjero	0. No 1. Si
21	**Clase de Cliente**	**0. Malo** **1. Bueno**

Fuente: Professor Dr. Hans Hofmann - Institut f"ur Statistik und "Okonometrie Universit"at Hamburg FB Wirtschafts wissenschaften Von-Melle-Park 5 2000 Hamburg 13.

El análisis a los datos se realizó bajo los tres métodos de clasificación, para lo cual se dividió la muestra en dos partes denominadas: muestra de entrenamiento (con 708 registros) y muestra de prueba (con 292 registros), se utilizaron las variables originales sin transformación, la selección de registros para las muestras fue completamente aleatoria.

4.4 Métodos de Clasificación

4.4.1 REGRESIÓN LOGÍSTICA

El método de Regresión Logística se aplicó con la finalidad de identificar las variables que influyen en los clientes que solicitan un préstamo. Para la estimación del modelo se procedió a utilizar solo la muestra de entrenamiento, obteniéndose:

$$p = \frac{1}{1+e^{-X'\beta}}$$

donde:

$$X'\beta = 0.441 - 0.037 * Duracion + 0.295 * Historial - 0.33 * Estado + 0.328 * EC\_Sexo + 0.475 * Telefono - 0.358 * Otros\_planes + 0.259 * \Pr opiedad - 0.42 * Alojamiento$$

Teniendo este modelo se procedió a evaluar su significancia considerando un nivel de significancia del 5%, resultando que dicho modelo es significativo (χ^2 = 96.1681, con 8gl y p_valor de 0.000). Respecto a la evaluación de la bondad de ajuste del modelo con un nivel de significancia del 5%, se obtuvo que el modelo ajustado es significativo (χ^2 = 757.693, con 789gl y p_valor de 0.2171). Además este modelo obtuvo una tasa de error del 0.2698, es decir, un 73.02% de buena clasificación lo cual indico que el modelo es bueno para clasificar.

Dado que es necesario conocer que variables son las que contribuyen en la clasificación del cliente, utilizando el método Forward para la selección de variables se obtiene que las variables: Estado de cuenta corriente existente (X_1), Duración en meses (X_2), Historial Crediticio (X_3), Estado civil y sexo (X_9), Propiedad (X_{12}), Otros planes de financiación (X_{14}), Alojamiento (X_{15}) y Teléfono (X_{19}) *son las que aportan más en el modelo.*

Las interpretaciones del Odds Ratio del modelo son:

- Las variables *"estado de cuenta corriente existente"*, *"duración del crédito en meses", "Otros planes de financiación" y "alojamiento"* favorecen el ser un buen cliente.
- *La posibilidad de encontrar un buen cliente es aproximadamente dos veces mayor a medida que el historial crediticio sea menor ó*

el estado civil y sexo varié ó si el banco cuenta con el teléfono del cliente ó si este posee bienes o inmuebles en relación a los que no poseen.

Para evaluar cuan bueno es para predecir este modelo, se utilizó la muestra de prueba y se obtuvo un 70.89% de buena clasificación lo cual indico que el modelo es bueno para predecir.

En la aplicación del análisis de regresión logística se realizó el análisis de residuos, obteniéndose lo siguiente:

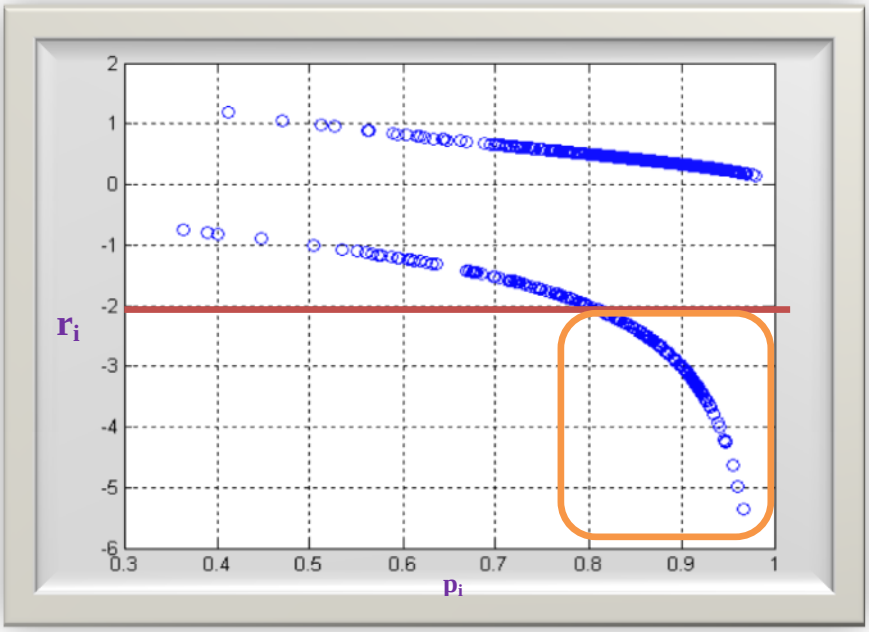


Figura IV.2: Gráfica p_i vs. r_i

De la gráfica de las probabilidades (p_i) y los residuos de Pearson (r_i) se aprecia un gran conjunto de valores discordantes (encerrados en el cuadro). En forma similar se aprecia en la figura IV.3, respecto a las probabilidades (p_i) y los residuos estudentizados (r^{est}_i).

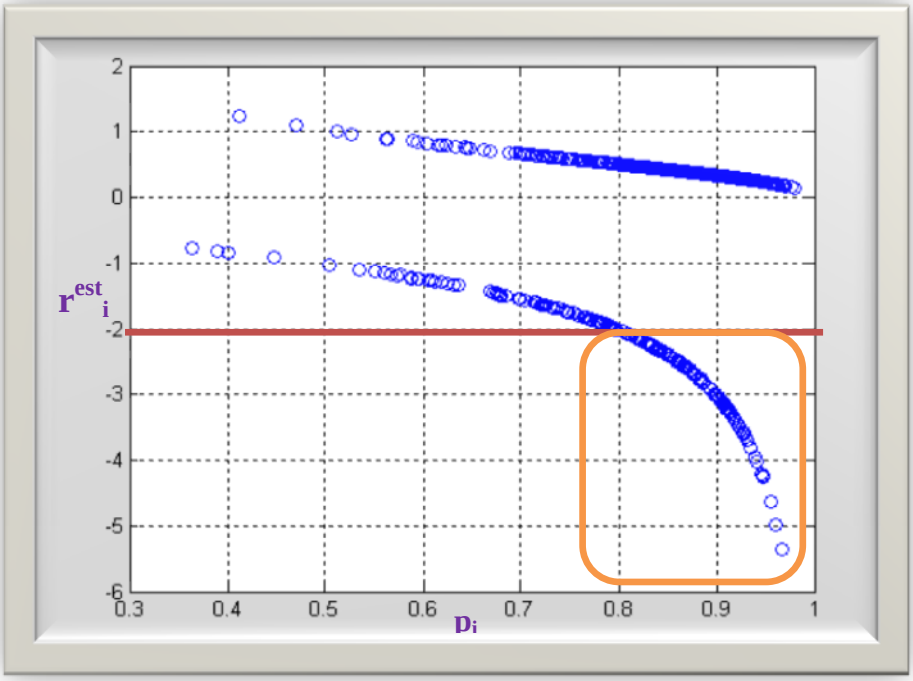


Figura IV.3: Gráfica p_i vs. r^{est}_i

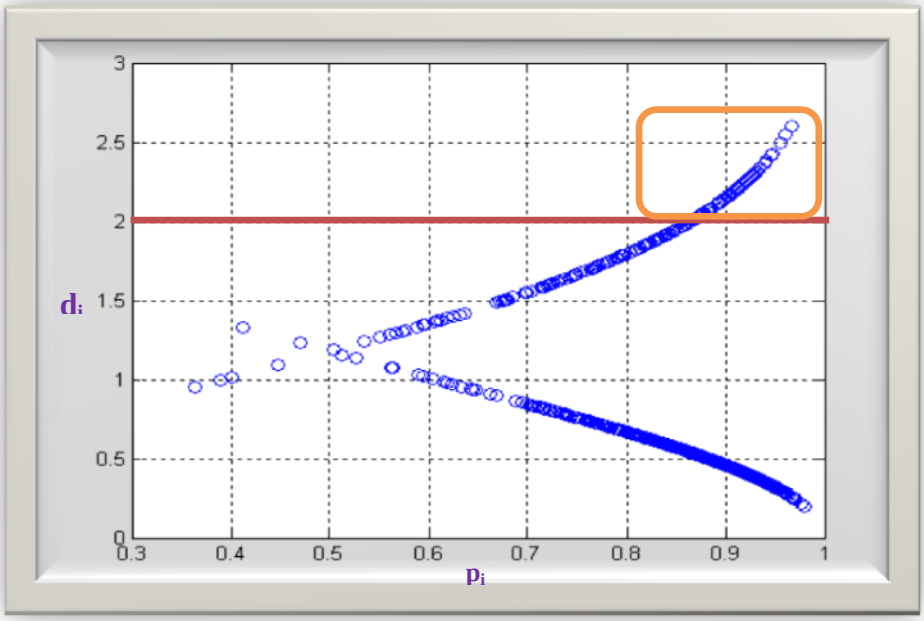


Figura IV.4: Gráfica p_i vs. d_i

De la gráfica de las probabilidades (p_i) y los residuos de Desviación (d_i) se aprecia un gran conjunto de valores discordantes (encerrados en el cuadro).

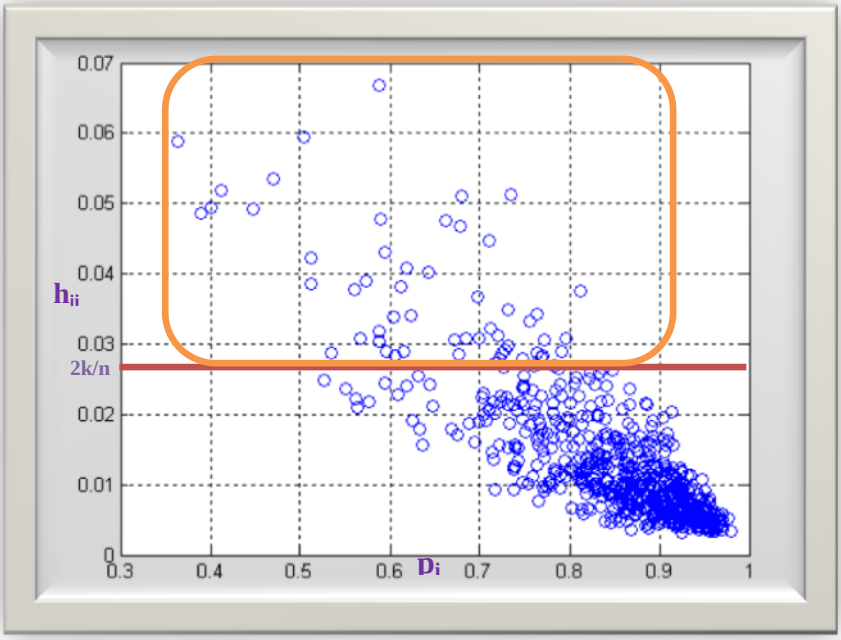


Figura IV.5: Gráfica p_i vs. h_{ii}

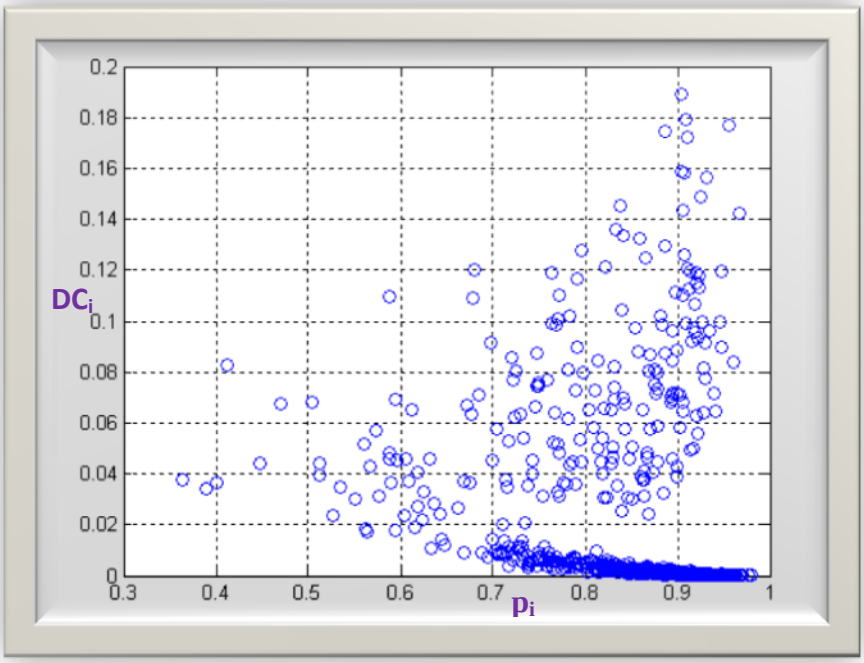


Figura IV.6: Gráfica p_i vs. DC_i

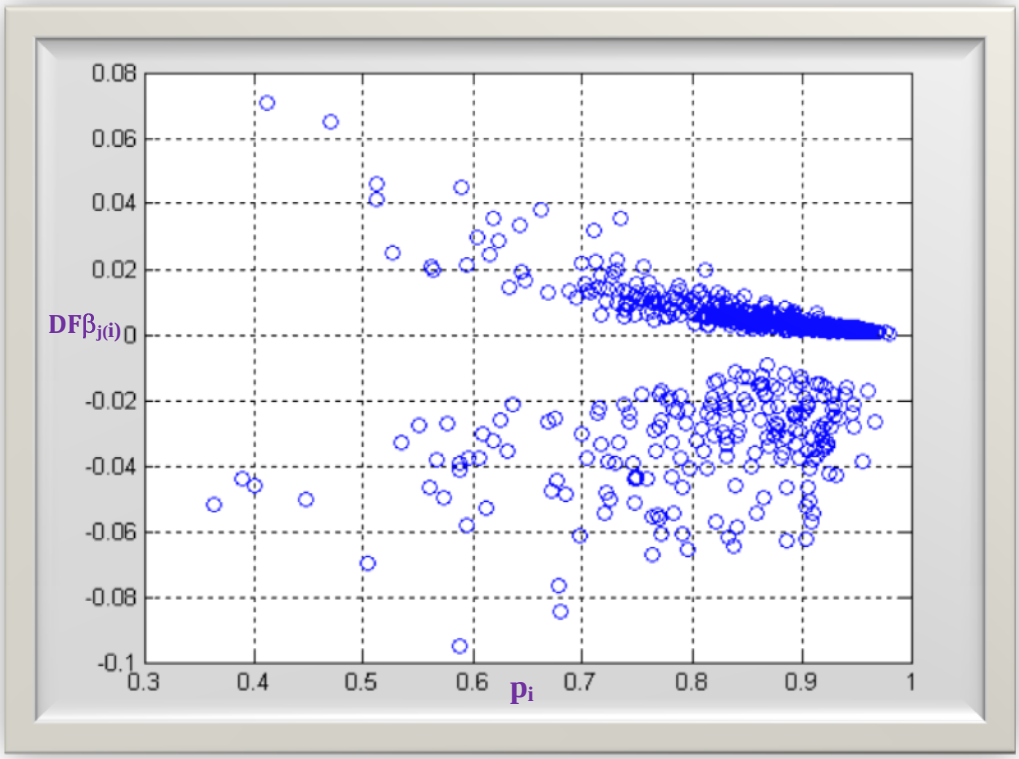


Figura IV.7: Gráfica p_i vs. $DF\beta_{j(i)}$

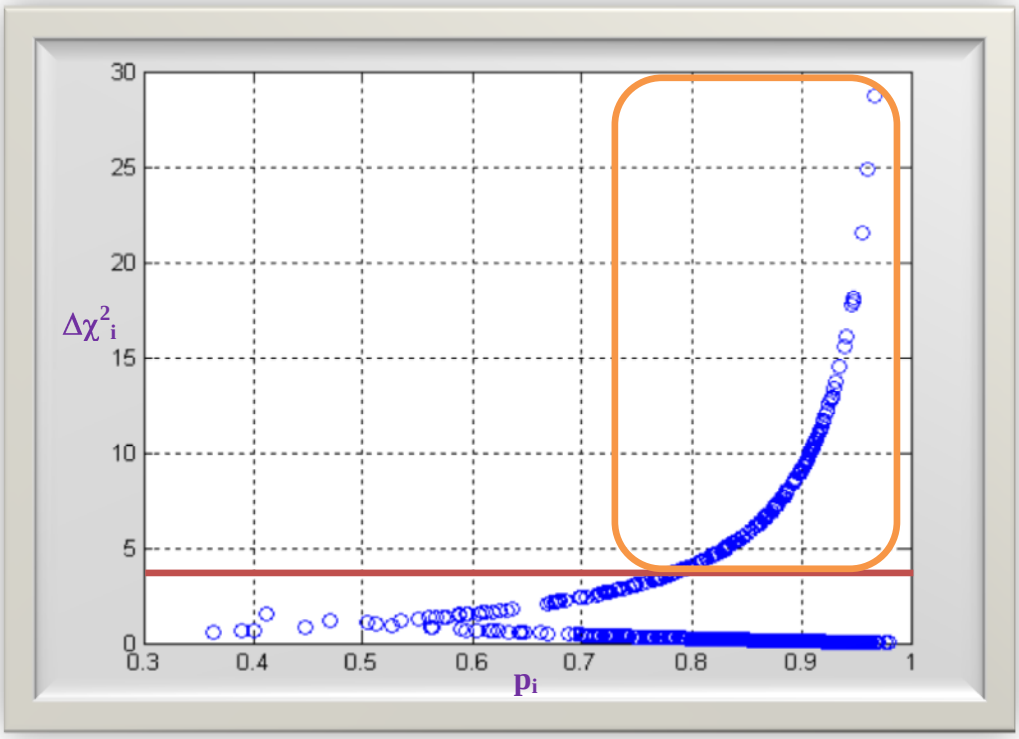


Figura IV.8: Gráfica p_i vs. $\Delta\chi^2_i$

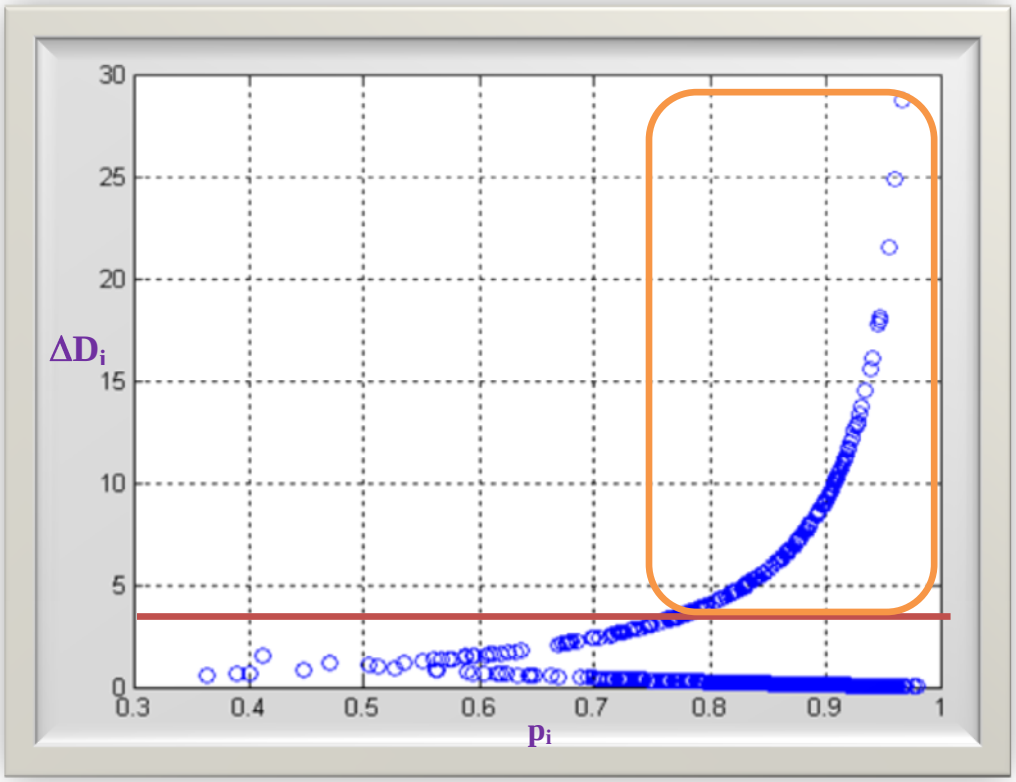


Figura IV.9: Gráfica p_i vs. ΔD_i

De la figura IV.5 de las probabilidades (p_i) y los leverage (h_{ii}) se aprecia un conjunto de valores influyentes superiores al valor 2k/n = 0.022599 (encerrados en el cuadro), sin embargo en la figura IV.6 de las probabilidades (p_i) y los valores de la distancia de cook (DC_i) no se aprecia valores influyentes, y en forma similar se aprecia en la figura IV.7, respecto a las probabilidades (p_i) y los valores de la DF Betas ($DF\beta_{j(i)}$).

De la figura IV.8 de las probabilidades (p_i) y valores de $\Delta\chi^2_i$ se aprecia un conjunto de valores influyentes superiores al valor 4 (encerrados en el cuadro), de forma similar se aprecia en la figura IV.9, respecto a las probabilidades (p_i) y los valores de ΔD_i se aprecia un conjunto de valores influyentes superiores al valor 4.

4.4.2 ÁRBOLES DE CLASIFICACIÓN (CART)

El método de Árboles de Clasificación (CART) se aplicó con la finalidad de identificar las variables que influyen en la clase de cliente que solicita un préstamo.

Para la estimación del árbol de clasificación se procedió a utilizar solo la muestra de entrenamiento, la cual fue particionada en: 474 registros para la construcción del árbol y 234 datos para la poda del este. Se considero como criterio de parada a la regla 1-SE, la cual se baso en la selección del mejor árbol podado, por lo que el mejor árbol posee siete nodos y cuatro hojas. Además este modelo obtuvo una tasa de error del 0.2542, es decir, un 74.58% de buena clasificación lo cual indico que el modelo es bueno para clasificar.

Las variables X_1, X_5 y X_3, son las que más favorecen en la clasificación de un cliente. Es decir:

- *Se obtendrá un buen cliente si este no tiene una cuenta corriente; si la tiene la cantidad del préstamo debe ser menor de 8150.5 DM y, tener una cuenta crítica o retraso en el pago.*

- *Un mal cliente será aquel que tenga una cuenta corriente, la cantidad del préstamo es menor de 8150.5 DM y, ser una persona que devuelve sus préstamos.*

Para evaluar cuan bueno es para predecir este modelo, se utilizó la muestra de prueba y se obtuvo un 72.60% de buena clasificación lo cual indico que el modelo es bueno para predecir.

4.4.3 REDES NEURONALES (PERCEPTRÓN MULTICAPA)

Aplicamos este método de Clasificación con la finalidad de conocer las variables que influyen en la clase de cliente que solicita un préstamo.

Para la estimación del modelo se procedió a utilizar solo la muestra de entrenamiento.

La red neuronal utilizada fue el Perceptron Multicapa con tres capas. Respecto a la clasificación de la red se tiene que según naturaleza es híbrida; según su arquitectura es multicapa con: una capa de entrada en donde se tuvo 20 neuronas correspondiente a las 20 variables que se tiene, una capa oculta en donde se tuvo 10 neuronas los cuales son combinaciones no lineales de las 20 variables y una capa de salida donde se tuvo 1 neurona, con dos posibles resultados mal pagador o buen pagador. El mecanismo de aprendizaje de esta red es supervisado. Según el tipo de asociación es una red unidireccional (feedforward), considerándose los siguientes parámetros de aprendizaje: Proporción del Conjunto de Validación 0.2, Tasa de Aprendizaje 0.15 y los datos fueron estandarizados. El algoritmo que se utilizó para la minimización de los errores es el de Backpropagation, con Regla de Parada igual a 100 (Máximo número de Iteraciones) y Tasa de Error del umbral 0.01.

Con respecto a la función de entrada esta red empleó la función lineal de hiperplano, la función de activación fue escalón, la función de salida es binaria.

Obteniéndose las siguientes características del aprendizaje de la red: 100 épocas, 0.1131 de Tasa de error del entrenamiento, 0.338 de Tasa de error de la validación y 56.3455 MSE del entrenamiento. Además este modelo obtuvo una tasa de error del 0.1582, es decir, un 84.18% de buena clasificación lo cual indico que el modelo es bueno para clasificar.

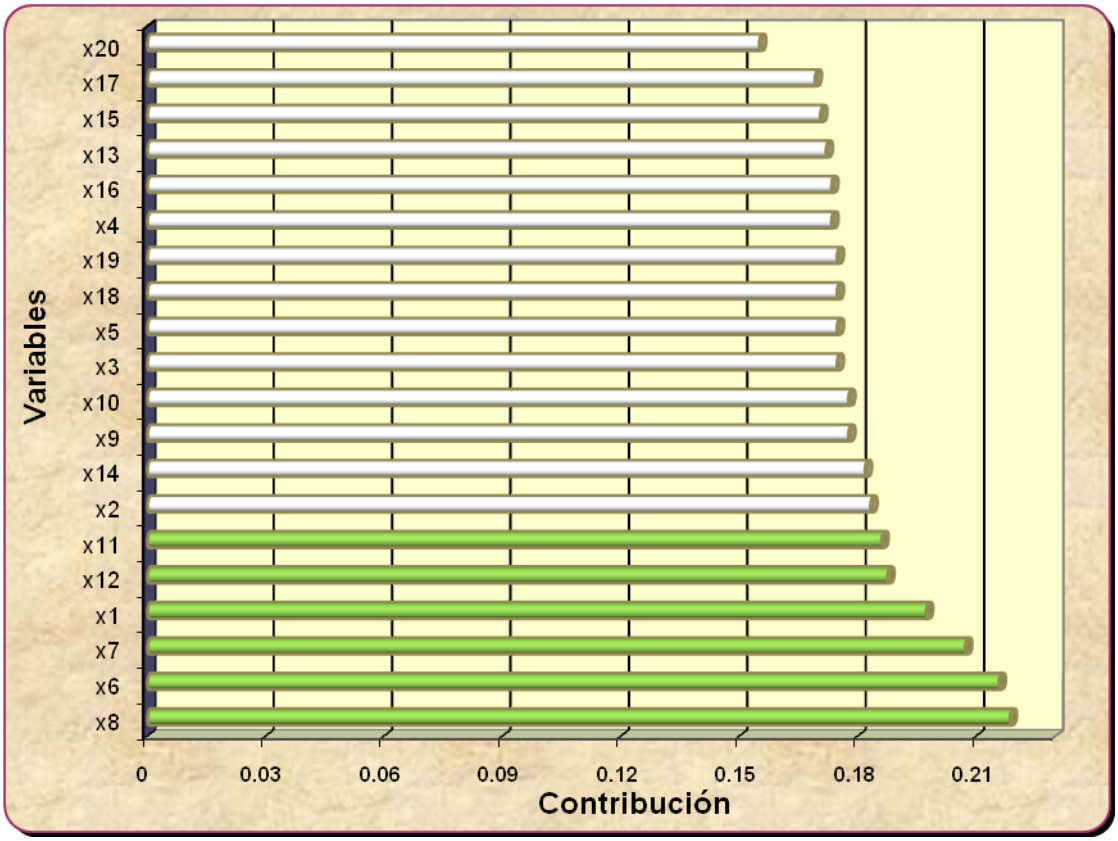


Figura IV.10: Contribución de las variables en la Red

Del gráfico se aprecio que las variables: Tasa de Amortización, Cuenta de Ahorros, Tiempo de Empleado, Estado de cuenta corriente, Propiedad y Tiempo de residencia contribuyen significativamente en el modelo de la Red Neuronal.

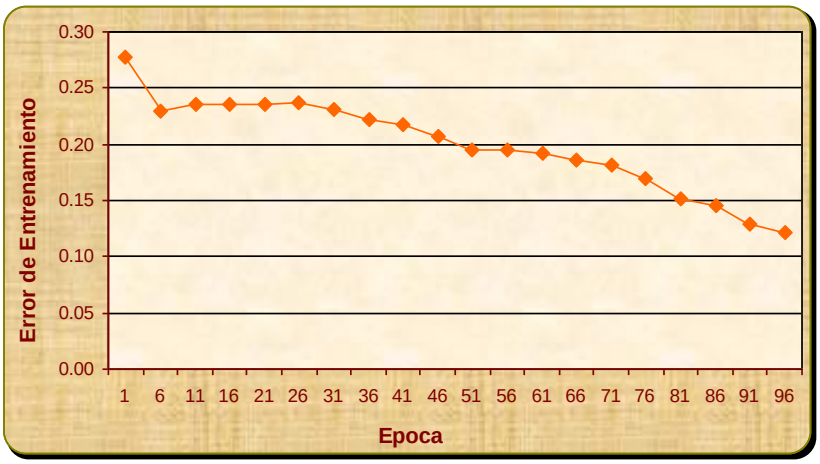


Figura IV.11: Error de Entrenamiento por Época

En la gráfica del Error de entrenamiento y de MSE se aprecia que estos han ido decreciendo a medida que sea mayor la época.

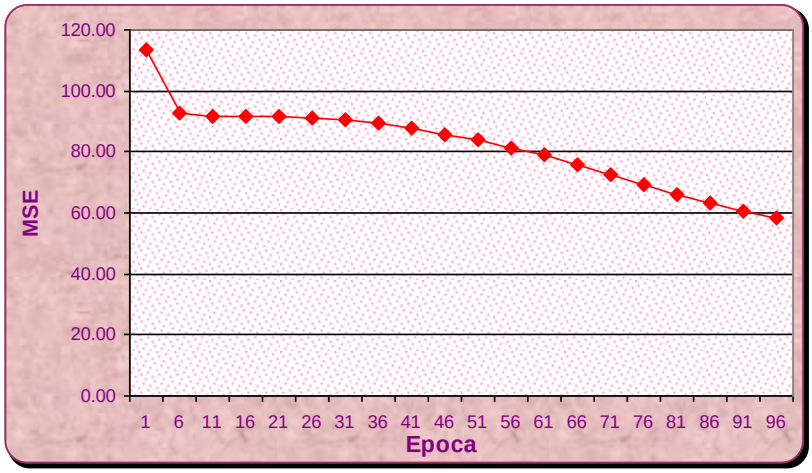


Figura IV.12: MSE por Época

Para evaluar cuan bueno es para predecir este modelo, se utilizó la muestra de prueba y se obtuvo un 74.32% de buena clasificación lo cual indico que el modelo es bueno para predecir.

TABLA RESUMEN

Nº	VARIABLES	Regresión Logística	CART	MLP
1	Estado de cuenta corriente existente	X	X	X
2	Duración del crédito en meses	X	-	-
3	Historial Crediticio	X	X	-
4	Objetivo del crédito	-	-	-
5	Cantidad de crédito	-	X	-
6	Cuenta de ahorros	-	-	X
7	Tiempo de Empleado	-	-	X
8	Tasa de amortización	-	-	X
9	Estado civil y sexo	X	-	-
10	Otros deudores ó garantes	-	-	-
11	Tiempo de Residencia	-	-	X
12	Propiedad	X	-	X
13	Edad	-	-	-
14	Otros planes de financiación	X	-	-
15	Alojamiento	X	-	-
16	Número de créditos en este banco	-	-	-
17	Trabajo	-	-	-
18	Número de personas que están obligadas a proporcionar mantenimiento	-	-	-
19	Teléfono	X	-	-
20	Trabajador Extranjero	-	-	-
21	**Clase de Cliente**	-	-	-
Nº DE VARIABLES		8	3	6
% de Error de Entrenamiento		0.2698	0.2542	0.1582
% de Error de Prueba		0.2911	0.274	0.2568
Comparando con la Regresión Logística	**Error Relativo de Entrenamiento**	100.00%	94.22%	58.64%
	% de Reducción		5.78%	41.36%
	Error Relativo de Prueba	100.00%	94.13%	88.22%
	% de Reducción		5.87%	11.78%

CONCLUSIONES Y RECOMENDACIONES

- No existe diferencias en el porcentaje de error de entrenamiento en los métodos: regresión logística y árbol de clasificación (CART), utilizados para la clasificación de los clientes que solicitan un préstamo; sin embargo con las redes neuronales se obtuvo un 84.18% de buena clasificación y un 74.32% de buena predicción.

- Con el modelo de Regresión Logística, se obtuvo mayor error en la clasificación y predicción debido a que este método es sensible a los valores influyentes, al igual que los árboles de clasificación (CART), por otro lado la red neuronal es insensible a valores influyentes.

- Para trabajos posteriores se recomienda utilizar meta modelos, que es la combinación de modelos diferentes, que subsanan el problema del sobreajuste de los datos de entrenamiento y que pueden mejorar la clasificación y la predicción.

BIBLIOGRAFÍA

[1]. Molinero, L. (2004). Historia del Razonamiento Estadístico. Publicado por Asociación de la Sociedad Española de Hipertensión y la Liga Española para la lucha contra la Hipertensión Arterial. Obtenido el 23/10/08 desde http://www.seh-lelha.org/historiastat.htm#LADY

[2]. Peña, D. (2002). Análisis de Datos Multivariantes. Publicado por Mc Graw Hill / Interamericana de España S.A.

[3]. Hosmer, D., Lemeshow, S. (1989) Applied Logistic Regression. Publicado por Ed.John Wiley, New York.

[4]. Caballero, J. (2008). Modelos de Regresión Logística Incondicional (II). Publicado por la Sociedad Andaluza de Enfermedades Infecciosas. Obtenido el 20/10/08 desde http://saei.org/hemero/epidemiol/nota5.html

[5]. Rojo, J. (2007) Regresión con variable dependiente cualitativa. Laboratorio de Estadística publicado por el Instituto de Economía y Geografía, Madrid. Obtenido el 08/07/09 desde http://www.cchs.csic.es/web_UAE/tutoriales/PDF/Regresion_variable_dependiente_dicotomica_3.pdf

[6]. Domínguez, E., Aldana, D. (2001) Regresión Logística. Revista Cubana Endocrinología 12(1): 58-64. Obtenido el 20/10/09 desde http://bvs.sld.cu/revistas/end/vol12_1_01/end07101.htm.

[7]. Acuña, E. (2004) Regresión con variables cualitativas. Universidad de Puerto Rico Recinto Universitario de Mayagüez. Obtenido el 03/02/09

desde http://math.uprm.edu/ ~edgar/cap5sl.ppt

[8]. Fernández, G. (2002). Data Mining Using SAS Applications. Publicado por Chapman & Hall / CRC.

[9]. Gámez, M., García, N. (2000) Rating de pequeñas y medianas empresas mediante árboles de clasificación. Publicado por la Universidad de Castilla – La Mancha. Obtenido el 20/10/08 desde http://www.uclm.es/ab/fcee/D_ tra bajos/2-2000-2.pdf.

[10]. Stacey Leung (2008). Análisis Comparativo entre Árboles de Clasificación, Publicado por Word Press.com. Obtenido el 18/01/09 desde http://techi322.wordpress.com/2008/04/16/analisis-comparativo-entre-arbo les-de-clasificacion-2/.

[11]. Acuña, E. (2004). Clasificación Usando Árboles de Decisión. Publicado por Universidad de Puerto Rico Recinto Universitario de Mayagüez. Obtenido el 16/07/09 desde http://math.uprm.edu/~edgar/clasifall9.pdf.

[12]. Morgan, J. and Messenger, R. (1973): THAID: a Sequential Search Program for the Analysis of Nominal Scale Dependent Variables, Ann Arbor: Institute for Social.

[13]. Puerta, A. (2002) IMPUTACIÓN BASADA EN ÁRBOLES DE CLASIFICACIÓN. Publicado por Eustat. Obtenido el 22/10/08 desde http:// www.eustat .es/document/datos/ct_04_c.pdf.

[14]. Marín. J. (2009). Análisis de Cluster y Árboles de Clasificación. Publicado por Universidad Carlos III de Madrid. Obtenido el 27/10/08 desde http:// halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema6dm.pdf.

[15]. Cortijo, F. (2001) Técnicas supervisadas II: Aproximación no paramétrica. Publicado por Computer Based Learning Unit, University of Leeds. Obtenido el 20/10/08 desde http://iie.fing.edu.uy/ense/asign/recpat/ material/tema3_ 00-01/node1.html.

[16]. McCulloch, W., Pitts, W. (1943) Un cálculo lógico de la inminente idea de la actividad nerviosa. Publicado en el boletín de matemática biofísica

5:115.133.

[17]. Rosenblatt, F. (1962) “Teorema de Convergencia del Perceptron”, Principios de Neurodinámica. Publicado por Spartan Books en New York.

[18]. Hilera, J., Martínez, V. (1995) Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones. Publicado por Ra-ma.

[19]. DARPA. (1988) Neural Network Study. Publicado por AFCEA International Press.

[20]. Freeman J. A. & Skapura D. M. (1991) Redes Neuronales. Algoritmos, aplicaciones y técnicas de programación. Publicado por Addison-Wesley Iberoamericana S.A.y Ed. Díaz de Santos S.A.

[21]. Haykin, S. (1994), Neural Networks: A Comprehensive Foundation. Publicado por MacMillan, en New York.

[22]. Nigrin, A. (1993), Neural Networks for Pattern Recognition. Publicado por Cambridge, MA: The MIT Press.

[23]. Kraneau, E. (2007) Separata: Fundamentos Teóricos de las Redes Neuronales.

[24]. Molera, L., Caballero, M. (2001) Predicción del Éxito en Estudios Universitarios mediante Redes Neuronales. Publicado en la X Jornada de la Asociación de Economía de la Educación en Murcia. Obtenido el 23/12/09 desde http://www.pagina-aede.org/Murcia/E07.pdf

[25]. Mtz. de Lejarza, I. (1998) Elementos Básicos de una Red Neuronal. Publicado por la Universidad de Valencia. Obtenido el 20/07/08 desde http:// www.uv.es/~mlejarza/ redes .htm

[26]. Piedra, N. (2007). Elemento Básicos de una Red Neuronal II, Publicado por Advanced Tech Computing Group UTPL, Loxa – Ecuador. Obtenido el 17/01/10 desde http://advancedtech.wordpress.com/2007/09/26/elementos

-baiscos-de-una-red-neuron al-artificialparte-ii/.

[27]. Serrano, C. Las redes neuronales artificiales. Publicado por 5Campus. Obtenido el 07/07/09 desde http://www.ciberconta.unizar.es/leccion/redes/INICIO.HTML

[28]. Serrano, C. Martin, B. (1995). Fundamentos de las redes neuronales artificiales: hardware y software. Obtenido el 17/03/10 desde http://es.wikipedia.org/wiki/Red_neuronal_artificial.

[29]. Galindo, P. (1999). Redes multicapa: Algoritmo Backpropagation. Publicado por la Universidad de Cádiz, España. Obtenido el 20/10/08 desde http://www2.uca.es/dept/leng_sist_informaticos/preal/23041/transpas/ E-Backpropagation/ppframe.htm.

[30]. MATLAB (2006). Tutorial de MATLAB R14 v7.0. Publicado por MATLAB.

[31]. Daza, S. Redes neuronales artificiales Fundamentos, modelos y aplicaciones. Publicado por la Universidad Militar Nueva Granada Facultad de Ingeniería Mecatrónica en Bogotá, Colombia. Obtenido el 20/10/08 desde http://www.monografias.com/trabajos12/redneur/redneur.shtml.

[32]. Hernández, J., Ramirez, M., Ferri, C. Introducción a la Minería de Datos. Publicado por Pearson Ed. Prentice Hall.

[33]. Palmer, A., Montaño, J., Jiménez, R. (2001) Tutorial sobre Redes Neuronales Artificiales. Publicado por la Universitat de les Illes Balears. Obtenido el 07/07/09 desde http://www.psiquiatria.com/psicologia/revista/61/2833.